



**ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO**

**FACULTAD DE MECÁNICA
ESCUELA DE INGENIERÍA DE MANTENIMIENTO**

**“DISEÑO E IMPLEMENTACIÓN DE DOS MÓDULOS PARA EL
CONTROL Y ENVÍO DE NOTIFICACIONES DE INGRESO DE
USUARIOS A LABORATORIOS EN TIEMPO REAL, CON
TARJETAS ARDUINO POR MEDIO DE LA RED GSM”.**

GRANIZO QUISIGUÍÑA CARINA ELIZABETH

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO DE MANTENIMIENTO

RIOBAMBA – ECUADOR

2015

E S P O C H

Facultad de Mecánica

C E R T I F I C A D O D E A P R O B A C I Ó N D E T E S I S

2013-12-20

Yo recomiendo que la Tesis preparada por:

G R A N I Z O Q U I S I G U I Ñ A C A R I N A E L I Z A B E T H

Titulada:

“DISEÑO E IMPLEMENTACIÓN DE DOS MÓDULOS PARA EL CONTROL
Y ENVÍO DE NOTIFICACIONES DE INGRESO DE USUARIOS A
LABORATORIOS EN TIEMPO REAL, CON TARJETAS ARDUINO
POR MEDIO DE LA RED GSM”

Sea aceptada como parcial complementación de los requerimientos para el Título de:

I N G E N I E R O D E M A N T E N I M I E N T O

Ing. Marco Santillán Gallegos.
DECANO DE LA FAC. DE MECÁNICA

Nosotros coincidimos con esta recomendación:

Ing. Ángel Silva Conde.
DIRECTOR DE TESIS

Ing. Marco Santillán Gallegos.
ASESOR DE TESIS

CERTIFICADO DE EXAMINACIÓN DE TESIS

NOMBRE DEL ESTUDIANTE: GRANIZO QUISIGUIÑA CARINA ELIZABETH

TÍTULO DE LA TESIS: "DISEÑO E IMPLEMENTACIÓN DE DOS MÓDULOS
PARA EL CONTROL Y ENVÍO DE NOTIFICACIONES DE INGRESO DE
USUARIOS. A LABORATORIOS EN TIEMPO REAL, CON TARJETAS
ARDUINO POR MEDIO DE LA RED GSM"

Fecha de Examinación: 2015-01-30

RESULTADO DE LA EXAMINACIÓN:

COMITÉ DE EXAMINACIÓN	APRUEBA	NO APRUEBA	FIRMA
Ing. Manuel González Puente PRESIDENTE TRIB. DEFENSA			
Ing. Ángel Silva Conde DIRECTOR DE TESIS			
Ing. Marco Santillán Gallegos ASESOR			

* Más que un voto de no aprobación es razón suficiente para la falla total.

RECOMENDACIONES:

El Presidente del Tribunal certifica que las condiciones de la defensa se han
cumplido.

Ing. Manuel González Puente
PRESIDENTE DEL TRIBUNAL

D E R E C H O S D E A U T O R Í A

El trabajo de grado presentado, es original y basado en el proceso de investigación y/o adaptación tecnológica establecido en la Facultad de Mecánica de la Escuela Superior Politécnica de Chimborazo. En tal virtud, los fundamentos teóricos - científicos y los resultados son de exclusiva responsabilidad de los autores. El patrimonio intelectual le pertenece a la Escuela Superior Politécnica de Chimborazo.

G r a n i z o Q u i s i g u i ñ a C a r i n a E l i z a b e t h

DEDICATORIA

Quiero dedicar este trabajo a mi familia, porque creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final. Va por ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí.

A mis tíos, primos y amigos, gracias por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida.

Mil palabras no bastarían para agradecerles su apoyo, su comprensión y sus consejos en los momentos difíciles. A todos, espero no defraudarlos y contar siempre con su valioso apoyo, sincero e incondicional.

Carina Elizabeth Granizo Quisiguiña

A G R A D E C I M I E N T O

En primer lugar quiero agradecer a Dios por cuidar cada uno de mis pasos y lograr culminar con éxito una etapa más como persona.

El más sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo, y en especial a la Escuela de Ingeniería de Mantenimiento, a los señores Ingenieros Ángel Silva y Marco Santillán director y asesor de mi trabajo de grado respectivamente, por brindarme la oportunidad de obtener una profesión y ser persona útil para la sociedad.

Y en especial para todos mis amigos, compañeros y personas que me apoyaron de una u otra manera para culminar con éxito una etapa de mi vida.

Carina Elizabeth Granizo Quisiguiña

CONTENIDO

		Pág.
1.	INTRODUCCIÓN	
1.1	Antecedentes	1
1.2	Justificación	2
1.3	Objetivos	2
1.3.1	<i>Objetivo general</i>	2
1.3.2	<i>Objetivos específicos</i>	2
2.	MARCO TEÓRICO	
2.1	Tarjetas de desarrollo de hardware y software libre	4
2.1.1	<i>Tarjetas de desarrollo de hardware libre</i>	4
2.1.2	<i>Tarjetas de desarrollo de software libre</i>	4
2.2	<i>Red GSM (Sistema global de comunicaciones móviles)</i>	4
2.3	<i>Tarjeta SIM (Módulo de identificación de abonado)</i>	4
2.4	<i>Red GPS (Sistema de posición global)</i>	5
2.5	<i>Red GPRS (Servicio general de paquetes vía radio)</i>	5
2.6	<i>Cable de datos USB 2.0</i>	5
2.7	<i>Arduino</i>	5
2.7.1	<i>Hardware Arduino</i>	6
2.7.2	<i>Escudo Arduino GSM</i>	6
2.7.3	<i>Tipos de tarjetas Arduino</i>	6
2.7.3.1	<i>Arduino Uno</i>	6
2.7.3.2	<i>Modulo Arduino Uno Normal</i>	6
2.7.3.3	<i>Arduino SMD</i>	7
2.7.3.4	<i>Shield V3.0</i>	7
2.7.3.5	<i>Arduino Leonardo</i>	8
2.7.3.6	<i>Arduino MEGA ADK</i>	8
2.7.3.7	<i>Arduino Micro</i>	8
2.7.3.8	<i>Arduino Nano</i>	9
2.7.3.9	<i>Arduino Ethernet</i>	9
2.7.3.10	<i>Arduino Due</i>	10
2.7.3.11	<i>Arduino BT</i>	10
2.8	<i>Partes básica del hardware Arduino</i>	10
2.8.1	<i>Conector USB</i>	11
2.8.2	<i>Regulador de voltaje de 5V</i>	11
2.8.3	<i>Plug de conexión para la alimentación externa</i>	11
2.8.4	<i>Puerto de conexiones</i>	11
2.8.5	<i>Puerto de entradas análogas</i>	12
2.8.6	<i>Microcontrolador Atmega 328</i>	12
2.8.7	<i>Botón de RESET</i>	12
2.8.8	<i>Pines de programación ICSP</i>	12
2.8.9	<i>LED ON</i>	12
2.8.10	<i>LEDs de recepción y transmisión</i>	13
2.8.11	<i>Puerto de conexiones de 7 pines</i>	13
2.8.12	<i>Puerto de conexiones de 5 pines</i>	13
2.9	<i>Software Arduino</i>	13
2.9.1	<i>Lenguaje de programación</i>	13
2.9.2	<i>Entorno de desarrollo en Arduino</i>	14
2.9.3	<i>Sketch o IDE</i>	15
2.9.3.1	<i>Función Verificar</i>	16
2.9.3.2	<i>Función Cargar</i>	16

2.9.3.3	<i>Función Abrir...</i>	16
2.9.3.4	<i>Función Guardar...</i>	16
2.9.3.5	<i>Función Monitor Serial...</i>	17
2.9.4	<i>Comandos AT...</i>	17
2.9.5	<i>Estructuras...</i>	18
2.9.5.1	<i>Estructuras Setup () y Loop...</i>	18
2.9.6	<i>Variables...</i>	18
2.9.6.1	<i>Variable Char (caracter)...</i>	18
2.9.6.2	<i>Variable int (entero corto)...</i>	19
2.9.6.3	<i>Variable long (entero largo)...</i>	19
2.9.6.4	<i>Variable boolean (Lógica)...</i>	19
2.9.7	<i>Constantes...</i>	19
2.9.7.1	<i>Constante HIGH LOW...</i>	19
2.9.7.2	<i>Constante input, output...</i>	19
2.9.8	<i>Funciones...</i>	20
2.9.8.1	<i>Pines digitales...</i>	20
2.9.8.2	<i>Pines analógicos...</i>	20
2.9.9	<i>Temporizadores...</i>	20
2.9.10	<i>Comunicación...</i>	20
2.9.11	<i>Estructuras de control...</i>	20
2.10	<i>Software LabVIEW...</i>	21
2.10.1	<i>Programación en el software LabVIEW...</i>	22
2.10.1.1	<i>Panel frontal y diagrama de bloques...</i>	22
2.10.1.2	<i>Menus de LabVIEW...</i>	23
2.10.1.3	<i>Herramientas de LabVIEW...</i>	24
2.11	<i>Sensores...</i>	25
2.11.1	<i>Sensor de movimiento infrarrojo (PIR)...</i>	25
2.11.2	<i>Sensor infrarrojo Sharp 2Y0A21...</i>	26
2.11.3	<i>Sensor de corriente análogo...</i>	26
2.11.4	<i>Sensor fotoeléctrico...</i>	27
2.11.5	<i>Sensor fotoeléctrico Encoders...</i>	28
2.11.6	<i>Sensor LM35...</i>	28
2.12	<i>Fundamentos de mantenimiento...</i>	29
2.12.1	<i>Objetivos del mantenimiento...</i>	29
2.12.2	<i>Tipos de mantenimiento...</i>	30
2.12.3	<i>Mantenimiento Productivo Total (TPM)...</i>	30
2.12.4	<i>Manuales de mantenimiento...</i>	31
2.12.4.1	<i>Objetivos del manual de mantenimiento...</i>	31
2.12.4.2	<i>Tipos de manuales de mantenimiento...</i>	31
3.	DISEÑO, PROGRAMACIÓN E IMPLEMENTACIÓN DE LOS MÓDULOS DE CONTROL DE INGRESO DE USUARIOS A LABORATORIO EN TIEMPO REAL Y TABLERO EQUIPADO CON VARIOS SENSORES UTILIZANDO SOFTWARE, HARDWARE ARDUINO Y LabVIEW.	
3.1	<i>Análisis del software y hardware Arduino...</i>	33
3.1.1	<i>Análisis del software Arduino...</i>	33
3.1.2	<i>Análisis del hardware Arduino...</i>	33
3.2	<i>Aplicación con software Arduino...</i>	34
3.2.1	<i>Programación que controla los módulos...</i>	35
3.2.1.1	<i>Primera parte...</i>	35
3.2.1.2	<i>Segunda parte...</i>	36
3.2.1.3	<i>Tercera parte...</i>	36
3.3	<i>Especificaciones técnicas de las tarjetas Shield y Arduino MEGA...</i>	38
3.3.1	<i>Módulo V3.0...</i>	38

3.3.2	<i>Arduino MEGA ADK...</i>	38
3.4	<i>Pruebas de funcionamiento del programa que controla los módulos...</i>	39
3.5	<i>Listado de los equipos, herramientas y materiales...</i>	39
3.6	<i>Implementación del módulo...</i>	40
3.6.1	<i>Montaje de los diferentes componentes en el módulo...</i>	40
3.6.2	<i>Montaje de las tarjetas Arduino: mega ADK y shield V3.0...</i>	44
3.6.3	<i>Conexiones de los elementos del módulo a las tarjetas Arduino...</i>	45
3.6.4	<i>Instalación del módulo en el laboratorio...</i>	47
3.7	<i>Tablero equipado con varios sensores controlado con LabVIEW - Arduino...</i>	48
3.7.1	<i>Aplicación con software LabVIEW...</i>	48
3.7.2	<i>Implementación del tablero con sensores...</i>	51
3.7.2.1	<i>Montaje de los sensores en el tablero...</i>	52
3.7.2.2	<i>Montaje de la tarjeta Arduino...</i>	52
3.7.2.3	<i>Adaptación de una tarjeta electrónica...</i>	52
4.	ELABORACIÓN DE MANUALES DE LOS MANUALES DE OPERACIÓN Y MANTENIMIENTO	
4.1	<i>Introducción...</i>	55
4.2	<i>Manual de operación de los módulos de control de ingreso de usuarios...</i>	55
4.3	<i>Elaboración del manual de mantenimiento de los módulos...</i>	56
4.3.1	<i>Fichas técnicas de los módulos adquiridos con sus respectivas partes...</i>	56
4.3.2	<i>Tareas de mantenimiento para los módulos implementados...</i>	57
4.3.3	<i>Banco y ejecución de las tareas de mantenimiento de los módulos...</i>	58
5.	CONCLUSIONES Y RECOMENDACIONES	
5.1	<i>Conclusiones...</i>	62
5.2	<i>Recomendaciones...</i>	64

BIBLIOGRAFÍA

ANEXOS

PLANOS

LISTA DE TABLAS

		Pág.
1	Partes básicas de un hardware Arduino	11
2	Características y aplicaciones del sensor de movimiento	25
3	Características y aplicaciones del sensor infrarrojo Sharp	26
4	Características y aplicaciones del sensor de corriente análogo	27
5	Características y aplicaciones del sensor fotoeléctrico	27
6	Características y aplicaciones del sensor Encoders	28
7	Características y aplicaciones del sensor LM 35	29
8	Especificaciones técnicas de la tarjeta Shield	38
9	Especificaciones técnicas de Arduino Mega	38
10	Listado de los equipos, herramientas y materiales	40
11	Ficha técnica de los módulos de control de ingreso	56
12	Ficha técnica de los módulos de control de ingreso	57
13	Banco de tareas para los módulos implementados en los laboratorios de electrónica de la Facultad de Mecánica	58
14	Ejecución de la tareas de mantenimiento, medición de voltaje de las tarjetas Arduino	59
15	Ejecución de las tareas de mantenimiento, revisión y limpieza externa del módulo de control	60
16	Ejecución de las tareas de mantenimiento, revisión y limpieza interna del módulo de control	61

LISTA DE FIGURAS

	P á g .
1 Logo	5
Ardui no	
2 Hardware Ardui no	6
3 Ardui no U no	6
4 Ardui no S M D	7
5 Microcontrolador AT mega	7
328	
6 Shield V3.0	7
7 Ardui no Leonardo	8
8 Ardui no MEGA ADK	8
9 Ardui no	9
Micro	
10 Ardui no Nano	9
11 Ardui no Ethernet	9
12 Ardui no Due	10
13 Ardui no BT	10
14 Partes básicas de un hardware Ardui no	11
15 Logo software	14
Ardui no	
16 Entorno de desarrollo en Ardui no	14
17 Función verificar	16
18 Función cargar	16
19 Función abrir	16
20 Función guardar	17
21 Función monitor serial	17
22 Logo LabVIEW Versión	21
2012	
23 Panel frontal	22
24 Diagrama de bloques	23
25 Menú de LabVIEW	23
26 Sensor de	
movim iento	25
27 Sensor Infrarrojo	
Sharp	26
28 Sensor de corriente análogo	26
29 Sensor fotoeléctrico	27
30 Sensor fotoeléctrico	28
Encoders	
31 Sensor LM35	28
32 Tipos de	30
mantenim iento	
33 Pantalla principal o	34
IDE	
34 Programación primera parte	35
35 Programación segunda	36
parte	
36 Código de usuario	36
37 Numero de	37
usuario	
38 Mensaje de	37

	confirmación	
39	Programación del tercera parte	37
40	Pruebas realizadas en el protoboard	39
41	Ubicación de la pantalla y teclado	41
42	Perforaciones para la pantalla y teclado	41
43	Perforaciones para el interruptor y potenciómetro	41
44	Orificio para el cable de alimentación	42
45	Protección para el cable	42
46	Montaje de la pantalla LCD y teclado lado posterior	42
47	Montaje de la pantalla LCD y teclado parte frontal	43
48	Montaje del interruptor ON/OFF y del potenciómetro	43
49	Montaje del regulador de voltaje	43
50	Tarjeta shield insertada en la tarjeta mega ADK	44
51	Unión de las tarjetas de las tarjetas en la base	44
52	Montaje pernos de soporte	45
53	Montaje de las tarjetas en los pernos de soporte	45
54	Conexión del teclado	45
55	Conexión de la pantalla LCD	46
56	Conexión del potenciómetro	46
57	Tarjeta electrónica	46
58	Módulo de control de ingresos	47
59	Realización de los agujeros para el empotrado del módulo	47
60	Empotrado del módulo	48
61	Módulo en funcionamiento	48
62	Diagrama de programación	49
63	Bloque de inicialización de conexión	50
64	Programación de lectura de datos análogos y digitales	51
65	Comando close	51
66	Montaje de sensores	52
67	Montaje de la tarjeta Arduino	52
68	Tarjeta electrónica en el tablero de sensores	53
69	Conexión de los diferentes sensores	53
70	Montaje de un motor de 9Vcc	53
71	Montaje de la batería y potenciómetro	54
72	Módulo de sensores	54

SIMBOLOGÍA

A	A m p e r i o s	A
V c a	C o r r i e n t e A l t e r n a	V
V c c	C o r r i e n t e C o n t i n u a	V
V	V o l t a j e	V

LISTA DE ABREVIACIONES

GSM	Sistema Global de Comunicación móviles
SIM	Módulo de Identificación de Abonado
GPS	Sistema de posición Global
GPRS	Servicio General de Paquetes Vía Radio
IDE	Entorno interactivo de desarrollo
FAME	Facultad de Mecánica
IP	Protocolo de Internet (Internet Protocol)
LABVIEW	Laboratorio de Instrumentación Virtual y Mesa de trabajo de Ingeniería (Laboratory Virtual Instrumentation Engineering Workbench)
LED	Diodo Emisor de Luz (Light Emitting Diode).
RAM	Memoria de Acceso Aleatorio (Random Access Memory)
RCM	Mantenimiento Centrado en la Confiabilidad (Reliability Centered Maintenance)
TPM	Mantenimiento Productivo Total (Total Productive Maintenance)
UNE	Una Norma Española
VI	Instrumento Virtual (Virtual Instrument)
LCD	Pantalla de Cristal Líquido (Liquid Crystal Display)
USB	Bus Universal en Serie (Universal Serial Bus)

LISTA DE ANEXOS

- A** Diagrama de procesos de operación del módulo de control
- B** Partes del panel de programación
- C** Ventana de programación en Arduino
- D** Partes básicas de un arduino
- E** Diagrama de bloques de operación del tablero de sensores
- F** Panel frontal del módulo de sensores

RESUMEN

Se ha realizado el diseño e implementación de dos módulos para el control y envío de notificaciones de ingreso de usuarios a laboratorios, en tiempo real, con software y hardware Arduino, con la ayuda de la red GSM la notificación de ingreso se recibirá en cualquier parte del territorio ecuatoriano, esto tiene como finalidad de mejorar el nivel de vigilancia en el interior de los laboratorios. Dichos módulos implementados se encuentran programados con la nueva tecnología de automatización existente en el país.

Se investigó como realizar la comunicación o interfaz de datos de la tecnología del software LabVIEW a una tarjeta electrónica Arduino Uno, logrando la implementación de un tablero equipado con sensores de varios tipos, la implementación de dicho tablero es de gran ayuda en desarrollo de prácticas en las cátedras de Electrónica, Automatización y Mantenimiento.

Para el desarrollo del manual de mantenimiento de los módulos de control de ingresos, se respetó el formato que fue establecido por la Facultad de Mecánica, es decir, que cuenta con aspectos como: área, ubicación y sistemas o equipos. Debido a que se tratan de equipos nuevos, para la determinación de las frecuencias de mantenimiento se tomó como referencia las proporcionadas por el manual del fabricante, no sin antes acotar que éstas son sólo una guía, ya que hay que recordar que "el modo de falla es aleatorio".

Es recomendable estar capacitado en los procesos de manejo y funcionamiento del equipo establecido en el manual de operación, seguridad y mantenimiento para poder disponer en largo tiempo del equipo que permite entrar a un nuevo proceso de enseñanza, aprendizaje de calidad.

ABSTRACT

A design and implementation of two modules for control and sending notifications entry users to laboratories has been made, in real time, with software and hardware Arduino, with the help of the GSM network admission notification will be received anywhere in the Ecuadorian territory, it aims to improve the level of any surveillance inside the laboratories. These modules implemented are programmed with the new existing technology of automation in the country.

It was investigated how to perform the communication or data interface LabVIEW software technology to an electronic card Arduino Uno, managing the implementation of this board is a great help in the development of practices in the departments of Electronics, Automation and Maintenance.

To develop the maintenance manual for entry users control modules the format was established by the faculty of Mechanical respected, which features aspects such as: area, location systems or equipment. Treated as new equipment, for determining maintenance frequencies was taken as reference those provided by the manufacturer's manual, remember that these are only a guide and "The failure mode is random".

It is recommended to be trained on management processes and operation to have in long time equipment that allows entering a new process of teaching, learning quality.

CAPÍTULO I

1. INTRODUCCIÓN

1.1 Antecedentes

La humanidad siempre ha tratado de facilitar el trabajo a través de la automatización. El nivel actual de la tecnología nos exige a seguir presentando nuevas formas de operar, manipular y mantener a todos los activos para lo cual fueron creados.

En nuestra vida diaria nos percatamos de que los procesos productivos en diversas ramas de la industria utilizan cada vez con mayor frecuencia equipos complejos compuestos por sistemas mecánicos, eléctricos, electrónicos, neumáticos e hidráulicos, los que interactúan entre sí para lograr que los sistemas productivos trabajen con mayor flexibilidad, versatilidad, seguridad y confiabilidad, así como un bajo consumo de energía.

La escuela de Ingeniería de Mantenimiento de la Facultad de Mecánica cuenta con el laboratorio de Electrónica, siendo de gran ayuda para los estudiantes permitiéndose complementar y consolidar los fundamentos teóricos en ilustraciones prácticas.

El laboratorio de Electrónica desde su implementación se construyó para la realización de prácticas de los estudiantes de las escuelas pertenecientes a la Facultad de Mecánica, pero con el transcurso del tiempo los módulos, sistemas y dispositivos electrónicos van quedando obsoletos y desactualizados, en resumen dichos elementos van cumpliendo su vida útil.

En la actualidad el laboratorio de Electrónica no está equipado con un sistema de seguridad, el cual será de gran ayuda instalar, para proteger lo que se encuentra en el interior, razón por lo cual se vio la necesidad de implementar la tecnología Arduino que será de valiosa ayuda para cuidar los activos que se encuentran en el interior, cada personal encargado ingresará al laboratorio por medio de códigos que se digitarán al entrar a él, esta codificación se lo destinará a personas como: docentes, conserje. etc. encargadas de este lugar.

1.2 Justificación

Actualmente el laboratorio de Electrónica no cuenta con un control de ingreso de usuarios, el cual es necesario para la protección de los activos que se encuentran en él, así, como la infraestructura en el interior. En consecuencia, surge la necesidad de instalar un sistema de ingreso de usuarios al laboratorio a través de módulo equipado con tecnología Arduino, Shield V3.0, adicional a esto se realizará un tablero didáctico con la comunicación Arduino LabVIEW para que los compañeros se familiaricen con la nueva tecnología.

El módulo de control de ingresos también generara notificaciones en tiempo real por medio de la red GSM, protegiendo los bienes dentro los laboratorios a demás proporciona un conocimiento del hardware y software de la tecnología actual, así como el adecuado manejo, manipulación y mantenimiento de los elementos que componen este sistema.

Por lo cual se plantea mediante la presente tesis de grado; El "Diseño e Implementación de dos módulos para el control y envío de notificaciones de ingreso de usuarios a laboratorios en tiempo real, con tarjetas Arduino por medio de la red GSM" para el laboratorio de Electrónica y el otro para el lugar que haga falta de la Facultad de Mecánica.

El presente trabajo de grado está enmarcado dentro de la quinta línea institucional de investigación de la Escuela Superior Politécnica de Chimborazo, literales d y e; y así cumplir con el objetivo número 10 del Plan Nacional del Buen Vivir 2013-2017.

1.3 Objetivos

1.3.1 Objetivo general. Diseñar e implementar dos módulos para el control y envío de notificaciones de ingreso de usuarios a laboratorios en tiempo real, con tarjetas Arduino por medio de la red GSM.

1.3.2 Objetivos específicos:

Desarrollar un programa en la plataforma de Arduino para interpretar y generar instrucciones GSM.

Configurar el módem GSM vía software para el control y monitoreo del acceso de usuarios a los laboratorios.

Implementar un tablero equipado con sensores de varios tipos y tarjetas Arduino con conexión a LabVIEW para el desarrollo de prácticas multipropósitos en Electrónica, automatización y mantenimiento.

Implementar circuitos para conexión de sensores a tarjetas de Arduino.

CAPÍTULO II

2. MARCO TEORICO

2.1 Tarjetas de desarrollo de hardware y software libre

Son de bajo costo, no se requieren licencias es decir son de uso legal, la programación es de alto nivel también posee una amplia comunidad de trabajo en todo el mundo y cuenta con diferentes módulos para desarrollar múltiples aplicaciones.(Libre, 2004)

2.1.1 Tarjetas de desarrollo de hardware libre. Son aquellas que crean diseños de aparatos informáticos de forma abierta, de manera que todas las personas puedan acceder, como mínimo a los planos de construcción de los dispositivos.

2.1.2 Tarjetas de desarrollo de software libre. Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades
- La libertad de distribuir copias.
- La libertad de mejorar el programa y hacer públicas.(Libre, 2004)

2.2 Red GSM (Sistema Global de Comunicaciones móviles)

Esta red permite las transmisiones digitales de voz y de datos, como mensajes de texto (SMS) o mensajes multimedia (MMS). Todo terminal móvil debe estar constituido por una tarjeta SIM y el propio dispositivo, normalmente un teléfono móvil.(Kioskea, 2011)

2.3 Tarjeta SIM (Módulo de identificación de abonado).

Es la encargada de identificar en la red al usuario y al terminal móvil.

2.4 Red GPS (Sistema de Posición Global)

Es un sistema de navegación por satélite que permite determinar en todo el mundo la posición de un objeto con una gran precisión. (GPS, 2009).

2.5 Red GPRS (Servicio general de paquetes vía radio)

Es un sistema de transmisión de voz y datos que se transportan mediante paquetes de radio, una tecnología avanzada con menor duración de tiempo y mayor calidad como por ejemplo tenemos GPS para coches, para personas etc. (GPS, 2009)

2.6 Cable de Datos USB 2.0

Se utiliza para verificar si el programa Arduino se encuentra bien ejecutado, debido que transfiere datos del programa a la placa a utilizarse. (Bricogeek, 2012)

2.7 Arduino

Arduino es una plataforma electrónica de hardware libre basada en una placa con un microcontrolador. Con software y hardware flexibles y fáciles de utilizar. Arduino ha sido diseñado para adaptarse a las necesidades de todo tipo de público, desde aficionados, hasta expertos en robótica o equipos electrónicos. También consta de un simple, pero completo, entorno de desarrollo, que nos permite interactuar con la plataforma de manera muy sencilla. Se puede definir por tanto como una sencilla herramienta de contribución a la creación de prototipos, entornos, u objetos interactivos destinados a proyectos multidisciplinarios y multitecnología. (Arduino, 2011)

Figura 1. Logo Arduino



Fuente: [http://osl.ulpgc.es/wosl/wp-content/uploads/2014/05/720px-](http://osl.ulpgc.es/wosl/wp-content/uploads/2014/05/720px-Arduino_Logo.svg.png)

[Arduino_Logo.svg.png](http://osl.ulpgc.es/wosl/wp-content/uploads/2014/05/720px-Arduino_Logo.svg.png)

2.7.1 Hardware Arduino. El hardware, en su mayoría, está constituido por una baquelita electrónica desarrollada en torno a un microcontrolador, estas placas se pueden adquirir ensambladas o construirlas directamente ya que en la página se encuentran los planos electrónicos y la licencia del producto lo permite.

Figura 2. Hardware Arduino



Fuente: <http://ingenieriaelectronica.org/imagenes/tiposarduino.png>

2.7.2 Escudo Arduino GSM. Permite que una placa Arduino se conecte a Internet, enviar y recibir SMS y hacer llamadas de voz utilizando la biblioteca de GSM. (Arduino, 2012)

2.7.3 Tipos de Tarjetas Arduino

2.7.3.1 Arduino Uno. Es el módulo más famoso de todas las placas Arduino, y se puede encontrar en dos versiones: El módulo Arduino uno normal y la versión SMD.

2.7.3.2 Módulo Arduino Uno Normal. Utiliza un microcontrolador ATMEGA328 en formato DIP (dual en el paquete de la línea), la Ventaja del formato DIP es que podemos extraer o reemplazar el microcontrolador por uno nuevo siempre que queramos. (Robótica, 2006)

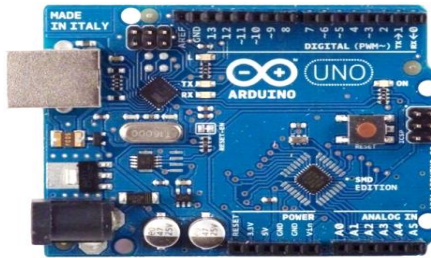
Figura 3.Arduino Uno



Fuente: <http://www.electan.com/Arduino-uno-con-atmega328-p-2977.html>

2.7.3.3 Arduino SMD. Esta tarjeta tiene un microcontrolador en formato SMD, es la única diferencia del módulo Arduino uno normal y va soldado en la superficie de la placa, pero sus funciones son similares. (Arduino, 2011)

Figura 4.Arduino SMD



Fuente: <http://Arduino.cc/en/uploads/Main/ArduinoUnoSmd.jpg>

- **Microcontrolador AT mega328.** Es un microcontrolador de la compañía Atmel que cuenta 32 KB de memoria flash, 2 KB de memoria RAM y 1KB de memoria EEPROM. (EMBAJADORES, 2010)

Figura 5. Microcontrolador AT mega328



Fuente: <http://www.electronicaembajadores.com/Productos/Detalle/-/1/LCA1ATMEGA328/arduino-atmega328-con-bootloader-uno>

2.7.3.4 Shield V3.0. La GPS/GPRS/GSM shield está basada en un módulo cuatribanda SIM908 que funciona en las frecuencias 850/900/1800/1900 MHz. También incorpora un

receptor GPS para recuperar datos de posicionamiento. La shield permite comunicarse con el mundo exterior por la red móvil, todo al mismo tiempo. (BricoGeek, 2012)

Figura 6. Shield V3.0



Fuente: <http://tienda.bricogeek.com/shields-arduino/572-gps-gprs-GSM-shield-v30.html>

2.7.3.5 Arduino Leonardo. Es el nuevo modelo del equipo de Arduino, utiliza un microcontrolador **ATMEGA32U4** que permite un diseño mucho más sencillo y económico. Una de las ventajas de este nuevo microcontrolador es que dispone de un puerto USB. También permite a la placa ser utilizada y programada como un dispositivo de entrada para imitar un teclado, ratón etc. (BricoGeek, 2012)

Figura 7. Arduino Leonardo



FUENTE: <http://www.bricogeek.com/shop/445-Arduino-leonardo.html>

2.7.3.6 Arduino MEGA ADK. Es una placa electrónica basada en el microprocesador Atmega2560 fabricada también por ATMEL, es conocido por muchos por ser el cerebro de las impresoras 3D, gracias a sus 54 pines de entrada o salida permite conectar un gran número de componentes a la placa. (SNOOTLAB, 2011)

Figura 8. Arduino MEGA ADK



FUENTE : <http://snootlab.com/lang-en/Arduino/296-Arduino-mega-adk-reV-3-en.html>

2.7.3.7 Arduino Micro. El nuevo Arduino Micro es una evolución de su hermano mayor Arduino Leonardo ya que utiliza el mismo chip ATmega32u4 pero en un formato mucho más pequeño de tan solo 48x18mm. Una de las mayores ventajas de este chip es que dispone de un puerto USB que permite entre otras cosas de evitar tener un conversor serie/USB y además de poder programar la placa como un dispositivo USB cliente como un teclado o un ratón por ejemplo. (Arduino, 2011).

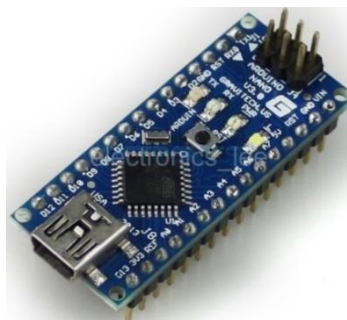
Figura 9. Arduino Micro



<http://Arduino.cc/en/uploads/Main/ArduinoMicroFront.jpg>

2.7.3.8 Arduino Nano. El Arduino Nano es una placa pequeña, completa es muy amistosa basada en el ATmega328. Tiene más o menos la misma funcionalidad del Arduino Duemilanove, pero en un paquete diferente. Le falta sólo un conector de alimentación de CC, y funciona con un cable USB Mini-B en vez de una normal. (Arduino, 2011).

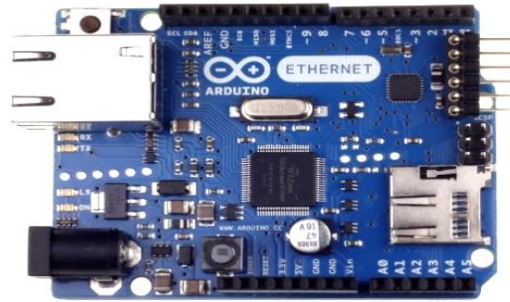
Figura 10. Arduino Nano



Fuente : <http://Arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano>

2.7.3.9 Arduino Ethernet.Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Está basada en el chip ethernetW iznetW 5100. El W iznet W 5100 provee de una pila de red IP capaz de TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas.(Arduino, 2012)

Figura 11.Arduino Ethernet



Fuente: <http://Arduino.cc/en/uploads/Main/ArduinoEthernetFront.jpg>

2.7.3.10 Arduino Due.Es un modelo más potente de las placas Arduino. Está basado en un potente microcontrolador SAM3x8E ARM corte v- M3 que incorpora todas las funcionalidades Clásicas de Arduino y añade otras nuevas(Arduino, 2011)

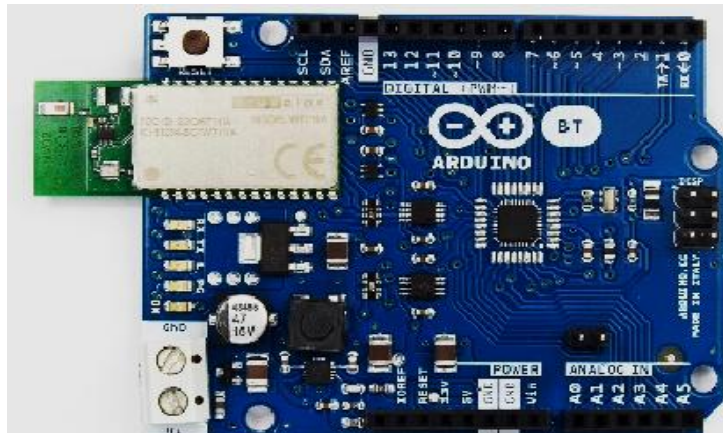
Figura 12.Arduino Due



Fuente: http://Arduino.cc/en/uploads/Main/ArduinoDue_Front.jpg

2.7.3.11 Arduino BT.contiene un módulo bluetooth que permite comunicar y programar sin necesidad de cableado.(Arduino, 2011).

Figura 13.El Arduino BT

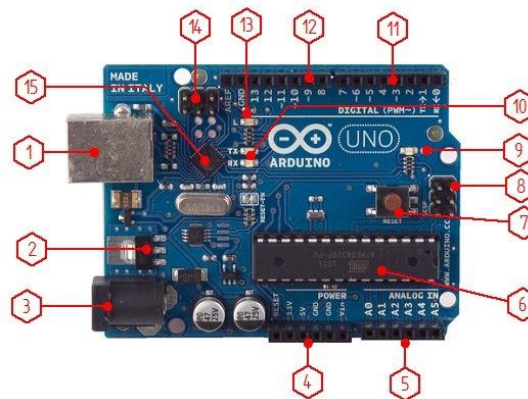


Fuente: http://Arduino.cc/en/uploads/Main/ArduinoBT_Front.jpg

2.8 Partes básicas del hardware Arduino.

El hardware Arduino consta de partes básicas como se muestran en la figura aunque no necesariamente todos los Arduinos son iguales, a continuación se describe a groso modo cada una de éstas.

Figura 14. Partes básicas de un hardware Arduino



Fuente: www.Arduino.cc

Tabla 1. Partes básicas de un hardware Arduino

1.- Conector USB.	9.- LED ON.
2.- Regulador de voltaje de 5V.	10.- LEDs de recepción y transmisión.
3.- Plug de conexión para la alimentación externa.	11.- Puerto de conexiones: está constituido por los pines de entradas o salidas digitales desde la cero hasta la 7.
4.- Puerto de conexiones.	12.- Puerto de conexiones: incluye 5 entradas o salidas adicionales (de la 8 a la 12).
5.- Puerto de entradas análogas.	13.- Este LED indica el estado del pin.
6.- Microcontrolador Atmega	14.- Estos pines se ocupan según la necesidad del usuario.

328.	
7.- Botón de RESET.	15.- Chip de comunicación que permite la conversión de serial a USB.
8.- Pines de programación ICSP.	

Fuente: Autora

2.8.1 Conector USB. Que puede ser tipo B o mini, este provee la comunicación para la programación y la toma de datos, también provee una fuente de 5V dc para alimentar al Arduino.

2.8.2 Regulador de voltaje de 5V. Se encarga de convertir el voltaje ingresado por el plug.

2.8.3 Plug de conexión para la alimentación externa. El voltaje que se suministra por aquí debe ser directo y estar entre 6V y 18V, incluso 20V, generalmente se usa un adaptador, pero debe tener cuidado de que el terminal del centro del plug quede conectado a positivo ya que algunos adaptadores traen la opción de intercambiar la polaridad de los cables.

2.8.4 Puerto de conexiones: constituido por 6 pines de conexión con las siguientes funciones:

- **RESET.-** permite resetear el microcontrolador al enviarle un cero lógico.
- **Pin 3.3V.-** este pin provee una fuente de 3.3Vdc para conectar dispositivos externos como en la protoboard por ejemplo.
- **Pin 5V.-** es una fuente de 5Vdc para conectar dispositivos externos.
- **Dos pines GND.-** que proveen la salida de cero voltios para dispositivos externos.
- **Pin Vin.-** este pin está conectado con el positivo del plug por lo que se usa para conectar la alimentación de la placa con una fuente externa de entre 6 y 12 Vdc en lugar del plug o la alimentación por el puerto USB. Este puerto esta modificado en la versión R3 de Arduino Uno.

2.8.5 Puerto de entradas análogas. Aquí se conectan las salidas de los sensores análogos. Estos pines solo funcionan como entradas recibiendo voltajes entre cero y cinco voltios directos.

2.8.6 Microcontrolador Atmega 328. Es el microcontrolador implementado en los Arduino uno y sobre el cual vamos a programar, en la versión SMD del Arduino uno R2, se usa el mismo microcontrolador pero en montaje superficial, en este caso las únicas ventajas que se me ocurren son la reducción del peso y ganar un poco de espacio.

2.8.7 Botón de RESET. Este botón así como el pin mencionado anteriormente permiten resetear el microcontrolador haciendo que reinicie el programa. En la versión R3 este pulsador se ubica arriba del conector USB, esto es un acierto pues al colocarle las Shield encima del Arduino, se perdía la opción de resetear dado que este pulsador quedaba tapado.

2.8.8 Pines de programación ICSP. Son usados para programar micro controladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.

2.8.9 LED ON. Indica que la tarjeta Arduino está encendido.

2.8.10 LED de recepción y transmisión. Éstos se encienden cuando la tarjeta se comunica con el PC.

- El Tx indica transmisión de datos.
- El Rx recepción.

2.8.11 Puerto de conexiones. Está constituido por los pines de entradas o salidas digitales desde la cero hasta la 7. La configuración como entrada o salida debe ser incluida en el programa.

Cuando se usa la terminal serial es conveniente no utilizar los pines cero (Rx) y uno (Tx). Los pines 3, 5 y 6 están precedidos por el símbolo ~ lo que indica, que permiten su uso como salidas controladas.

2.8.12 Puerto de conexiones. Incluye 5 entradas o salidas adicionales (de la 8 a la 12), las salidas 9, 10 y 11 permiten control por ancho de pulso; la salida 13 es un poco diferente pues tiene conectada una resistencia en serie, lo que permite conectar un LED directamente entre ella y tierra. Finalmente hay una salida a tierra GND y un pin AREF que permite ser empleado como referencia para las entradas analógicas. (Diseño, 2011).

También tenemos otros elementos como:

- **LED indicador de estado.** Este LED indica el estado del pin 13.
- **Pines de necesidad del usuario.** Estos pines se ocupan según la necesidad del usuario.

2.9 Software Arduino

2.9.1 Lenguaje de programación. La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel processing. Sin embargo es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino como: Java, Flash (mediante Action Script). Processing, Pure Data. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación del mismo, basado en Wiring y el entorno de desarrollo Arduino basado en Processing. (Corto, 2010)

Figura 15 Logo software Arduino



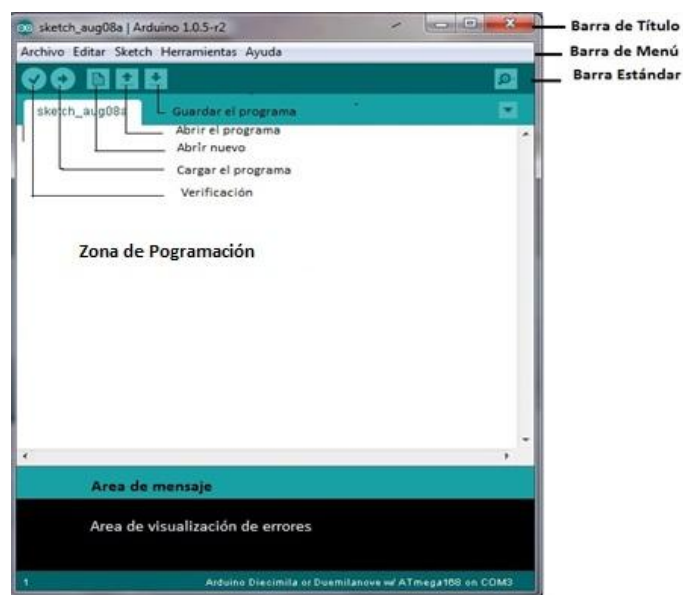
Fuente: www.arduino.cc.

- **Wiring.-** Es una plataforma abierta de propósitos electrónicos compuesta de un entorno de programación, una tarjeta de prototipo, documentación creada por ingenieros es decir por expertos, personas comparten su idea, conocimientos de forma colectiva.

- *Processing*.- Es un lenguaje de programación de código abierto, para las personas que quieran crear imágenes, animaciones e interacciones.

2.9.2 Entorno de desarrollo en Arduino. El entorno de desarrollo en Arduino (IDE) es el encargado de la gestión de la conexión entre el PC y el hardware de Arduino con el fin de establecer una comunicación entre ellos por medio de la carga de programas. Como podemos ver en la figura, el IDE de Arduino se compone de: (DOMÍNGUEZ, 2000).

Figura 16. Entorno de desarrollo en Arduino



Fuente: www.arduino.cc.

A continuación se describe las partes básicas del entorno de desarrollo:

- Un editor de texto.- donde escribir el código del programa.
- Un área de mensajes.- a través del cual el usuario tendrá constancia en todo momento de los procesos que se encuentren en ejecución, errores en código, problemas de comunicación, etc.
-
- Una consola de texto. Mediante la que podremos comunicarnos con el hardware Arduino y viceversa.

- Una barra de herramientas. Donde podremos acceder a una serie de menús y a los botones con acceso directo a las principales funcionalidades de Arduino.

A través de la IDE de Arduino, podemos escribir el código del programa software y crear lo que se conoce por "sketch" (programa).

2.9.3 Sketch o IDE. El IDE de Arduino viene en lenguaje de programación "Processing", enfocado al mundo gráfico, cada código es considerado un boceto, en inglés "sketch".

El sketch permite la comunicación con la placa Arduino. Estos programas son escritos en el editor de texto, el cual admite las posibilidades de cortar, pegar, buscar y reemplazar texto. En el área de mensajes se muestra, tanto la información mientras se cargan los programas, como los posibles errores que tengamos a la hora de compilar, ya sea por problemas en el código del sketch, por fallo en la detección de nuestro Arduino en el puerto USB, o por cualquier otro problema que sea detectado.

Desde la barra de herramientas tenemos acceso directo a las principales funciones que ofrece el IDE de Arduino, como por ejemplo: verificar el proceso de carga, crear un nuevo sketch, abrir un sketch ya existente, guardar los programas, abrir el Monitor Serial, etc.

A continuación se describe la utilidad de cada uno de los iconos que aparecen en la pantalla principal del entorno de desarrollo de Arduino:

2.9.3.1 Función verificar. Esta funcionalidad se encarga de verificar el código del sketch en busca de posibles errores.

A través del área de mensajes se le notificará al usuario el resultado de dicha verificación. En el caso de que se detecten errores en el código, éstos se detallarán junto con el número de línea en la que han sido detectados. Sólo cuando la comprobación resulta libre de errores podremos proceder a la carga del código en nuestra placa Arduino. (DOMÍNGUEZ, 2000)

Figura 17. Función verificar.



Fuente: www.arduino.cc

2.9.3.2 Función cargar. Permite compilar el código del sketch y lo carga en Arduino. Cuando la carga a terminado se informa al usuario a través del área de mensajes, y podremos proceder a la apertura del monitor serial.

Figura 18.Función cargar.



Fuente: www.arduino.cc

2.9.3.3 Función abrir. Permite abrir un sketch ya existente que ha sido previamente guardado. También puedes abrir cualquiera de los sketches que trae instalados por defecto el IDE de Arduino.

Figura 19.Función abrir.



Fuente: www.arduino.cc

2.9.3.4 Función guardar. Esta funcionalidad nos permite almacenar el sketch que estemos desarrollando en ese momento. Te permite elegir la ruta en la que quieres guardarlo, y te crea automáticamente una carpeta con el mismo nombre que le des al sketch, guardando éste dentro de la misma.

Figura 20.Función guardar.



Fuente: www.arduino.cc

2.9.3.5 Función monitor serial. Al pinchar sobre este icono, el entorno de desarrollo de Arduino abre una nuevaventana a través de la cual podemos ver la comunicación

establecida por el puerto serie entre la placa Arduino y el PC durante la ejecución del programa.

Figura 21. Función monitor serial.



Fuente: www.arduino.cc

2.9.4 Comandos AT. Los comandos AT, también conocidos como comandos Hayes (en honor a su desarrollador Dennis Hayes), Son una serie de instrucciones que establecen un interfaz de comunicación entre el usuario y modem, Su abreviatura AT por la que son mundialmente conocidos estos comandos proviene de la palabra "attention".

Aunque la finalidad principal de los comandos AT fue la comunicación con módems, la telefonía móvil GSM/GPRS también adoptó este lenguaje como estándar de comunicación.

En la actualidad, todos los terminales móviles GSM poseen una serie específica de comandos AT que nos permiten configurarlos por medio de estas instrucciones e indicarles una serie de acciones que se debe ejecutar, como marcar un número de teléfono, enviar o leer un SMS, consultar al estado de conexión a la red, leer o escribir en la agenda de contactos, etc.

Gracias a que la transmisión de comandos AT no depende del canal de comunicación a través del cual estos sean enviados (cable, infrarrojos, Bluetooth, etc.), podremos utilizar nuestra placa Arduino para transmitir dichos comandos a un módulo GPRS/GSM que sea capaz de interpretarlos y actuar en consecuencia.

A continuación, se va a dar a conocer los comandos AT que son de principal interés en el desarrollo del proyecto.

AT: Con este comando verificaremos que el módulo está recibiendo las instrucciones. Si todo es correcto, tras enviarlo debe responder con un "OK". En caso contrario no tendremos respuesta alguna.

`AT+CMGF=` ☐ `f` ☐ Selecciona el formato del mensaje SMS, donde:

☐ `f` ☐ = 0 modo PDU.

☐ `f` ☐ = 1 modo texto.

`AT+CMGS="num_móvil"`. A través de este comando marcaremos el número del móvil al que queremos hacer llegar el SMS. Irá delimitado por comillas.

2.9.5 Estructuras

2.9.5.1 *Arduino tiene dos estructuras: Setup () y Loop.*

- *Setup ()*. Es llamada justo en el momento en que el programa comienza. Se utiliza para inicializar variables y asignar los valores iniciales.
- *Loop*. Es la estructura que repite continuamente, permitiendo que el programa cambie y responda. Se usa para controlar de forma activa la tarjeta Arduino.

El tiempo varía según el número de instrucciones que contenga, y se puede conocer y controlar con las funciones de temporización como: `millis ()`, `delay ()`.

2.9.6 Variables

2.9.6.1 *Variable Char (carácter)*. Clasifica el tipo de datos para definir caracteres (ASCII), simbología tipo gráficos tales como A, d y \$.

Cada tipo char ocupa un byte (8 bits) de memoria y debe ser delimitado por comillas sencillas.

- `char var`
- `char var = "val"`

2.9.6.2 *Variable int (entero corto)*. El tipo entero, es un tipo de datos que almacena números y los almacena en campos de 2 byte (16 bits).

- `int LEDPin = 13`

2.9.6.3 Variable long (entero largo). El tipo de entero largo almacena un número de rango extendido y los almacena en campos de 4 byte (32 bits).

2.9.6.4 Variable boolean (lógica). Tipo de datos para valores booleanos de verdadero (true) o falso (false).

Es común usar valores de tipo boolean con la estructura de control y para determinar el flujo o secuenciación de programa.

- `boolean Var = true`

2.9.7 Constantes. Las constantes son variables predefinidas en el sistema. Son usadas para hacer que los programas sean más fáciles de leer.

2.9.7.1 Constante HIGH, LOW. Cuando se realiza la lectura o escritura sobre un pin digital, sólo hay dos valores posibles que el pin puede tomar o que se le puede asignar: HIGH, LOW.

- `high` = equivale a 5 V.
- `low` = Equivale a 0 V.

2.9.7.2 Constante input i output. Los pines digitales pueden ser usados tanto como en modo input (entrada) o modo output (salida).

2.9.8 Funciones

2.9.8.1 Pines digitales

- `PinMode (pin, mode)`. Configura el pin especificado para que se comporte como una entrada(input) o salida (output).
- `DigitalWrite (pin, value)`. Asigna el valor de salida high o low al pin especificado.
- `int digitalRead (pin)`. Lee o capture el valor de entrada del pin específico, dará valores high o low.

2.9.8.2 Pines analógicos

- *IntanalogRead (pin)*. Lee o captura el valor de entrada del específico pin analógico, la tarjeta Arduino realiza una conversión analógica a digital de 10 bits.

2.9.9 Temporizadores

- *Delay (ms)*. Detiene el programa durante una cantidad de tiempo (en milisegundos) especificado mediante parámetro.
- *Delaymicroseconds (milisegundos)*

2.9.10 Comunicación

- *Serial*. Se usa para crear la comunicación entre la placa Arduino y una computadora u otros dispositivos. Se declara en el *voidsetup()*

2.9.11 Estructuras de control

- *if (condición)*. La estructura if comprueba si la condición contenida entre paréntesis () se ha cumplido, como por ejemplo si una entrada supera un cierto número.
- *If... .else*. Esta estructura da mayor control sobre el flujo o secuenciación del código que la estructura de control básica if, permitiendo agrupar múltiples comprobaciones juntas.
- *For*. Realiza el control sobre una secuencia de repetición.
- *Switch case*. Ayuda en el control del flujo o secuenciación de los programas.
- *While*. Realiza un bucle de forma continua hasta que la expresión contenida dentro de los paréntesis (9 deja de ser Verdad).

- **Break.** Se utiliza para salir sin pasar por el estado normal del programa cuando este se repita. También se usa para salir de la función de interruptor comunicado.

2.10 Software LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench). Es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control, permite diseñar interfaces con el usuario mediante una ventana interactiva basado en software. Se puede diseñar especificando un sistema funcional, el diagrama de bloques o una notación de diseño de ingeniería. LabVIEW es a la vez compatible con herramientas de desarrollo similares. Tiene la ventaja de que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos. Por tanto, cuando se ejecuta un programa que funciona como instrumento virtual (VI), el usuario ve en la pantalla de su computador un panel cuya función es idéntica a la de un instrumento físico, facilitando la visualización y control del equipo. (NATIONAL, 2012)

Figura 22. Logo LabVIEW Versión 2012.



Fuente: National Instruments

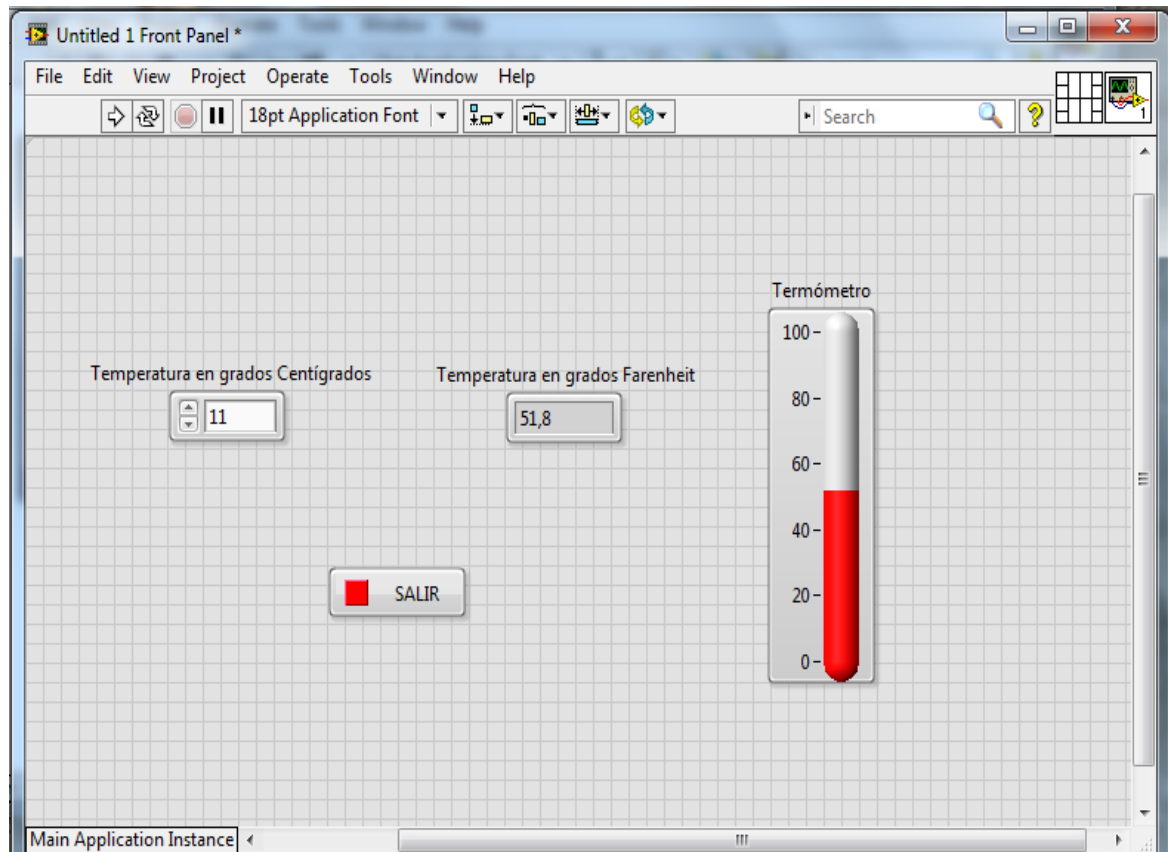
2.10.1 Programación en el software LabVIEW. Cuando se crea un instrumento virtual en LabVIEW se trabaja en dos ventanas.

Si un control es pegado desde la librería en el panel frontal, se crea una variable cuyos valores serán determinados por el usuario; inmediatamente aparecerá un terminal en la ventana de programación.

2.10.1.1 Panel frontal y diagrama de bloques. Cada VI tiene dos ventanas, pero relacionadas entre sí como se muestra a continuación.

En el panel frontal se elige controles e indicadores, éstos están asociados a cualquier tipo de datos que se hayan programado.

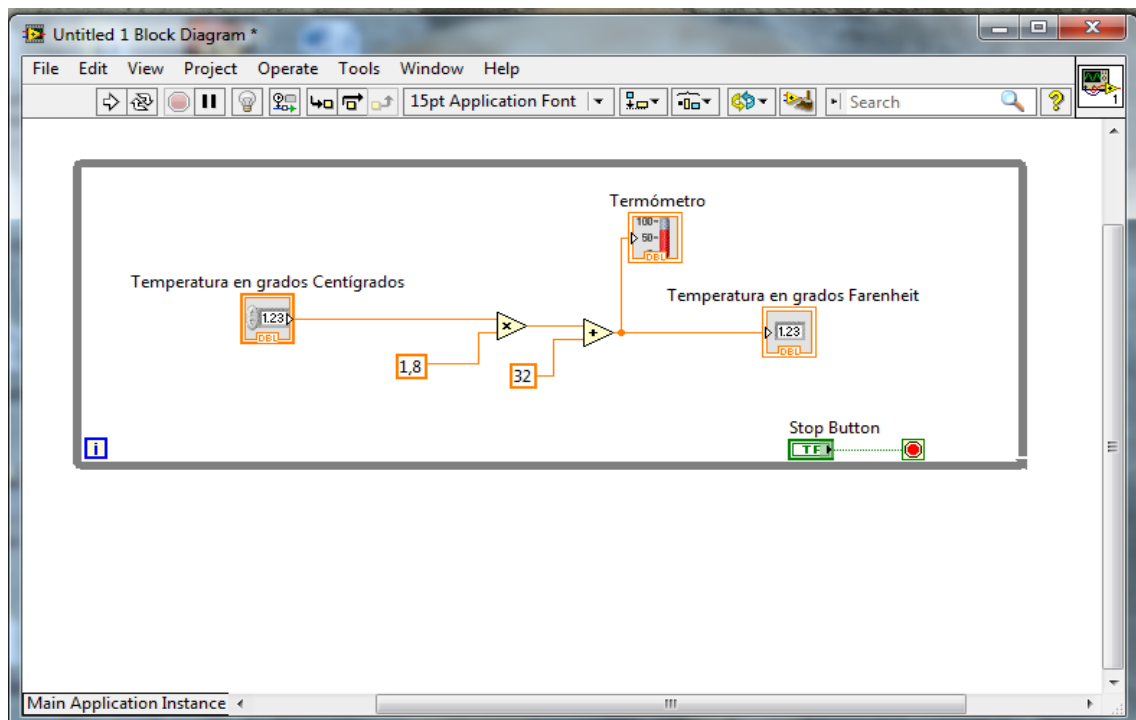
Figura 23. Panel frontal.



Fuente: Autora.

En la ventana diagrama de bloques se construye el programa que controla el equipo, se denomina también ventana de programación ésta ventana se la puede comparar con la placa de un circuito impreso, donde los terminales del panel frontal simulan un cableado con bloques funcionales.

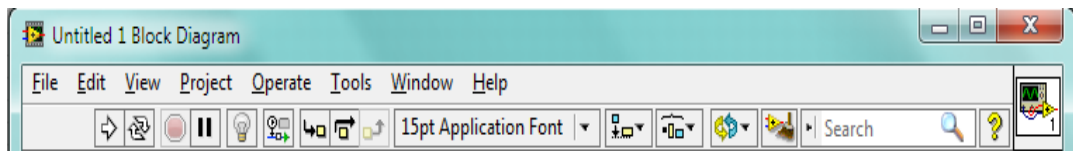
Figura 24. Diagrama de bloques.



Fuente: Autora.

2.10.1.2 Menús de LabVIEW. En la figura se muestra el menú de LabVIEW.

Figura 25. Menú de LabVIEW.



Fuente: National Instruments

- File: Sus opciones se usan básicamente para abrir, cerrar, guardar e imprimir VIs.
- Edit: Se usa principalmente para organizar el panel frontal y el diagrama de bloques y establecer nuestras preferencias.
- View: con esta opción encontramos la paleta de trabajo y lista de errores.
- Project: Presenta los niveles de jerarquía, de los sub-VIs que lo integran, también busca VIs que están sin abrir, etc.
- Operate: Son comandos que sirven para ejecutar el VI.

- **Tools:** son las herramientas con las cuales vamos a vincular el LabVIEW con otros dispositivos como la tarjeta DAQ.
- **Windows:** Se utiliza para mostrar ventanas como: las de controles y funciones, herramientas, portapapeles, historial, etc. Si se selecciona mostrar jerarquía del VIs (Show).

2.10.1.3 Herramientas. En este menú se muestran las herramientas básicas que se usan en LabVIEW.

- **Valor operativo:** Maneja los controles del panel frontal. Es la única herramienta disponible en el modo RUN.
- **Situación/tamaño/selección:** Selecciona mueve y redimensiona objetos.
- **Edición de texto:** Crea y edita textos.
- **Conexión de cables:** Enlaza objetos del diagrama de bloques y asigna a los terminales del conector del VI los controles e indicadores del panel frontal.
- **Menú pop-up del objeto:** Despliega el menú Pop-Up asociado al objeto.
- **Desplazamiento de la pantalla:** Desplaza la pantalla en la dirección que deseamos para ver posibles zonas ocultas.
- **Establecer-quitar punto de ruptura:** Permite poner tantos puntos de ruptura como deseamos a lo largo del diagrama de bloques. Usamos esta misma herramienta para eliminar los puntos.
- **Sonda de datos:** Funciona con la opción *Prove*, sirve para probar los valores intermedios dentro de un VI.
- **Capturar color:** Permite saber de manera específica que color tiene un objeto, texto u otros elementos.
- **Colorear:** Colorea diversos objetos.

2.11 SENSORES

2.11.1 Sensor de movimiento Infrarrojo (PIR). El sensor PIR es un dispositivo piroeléctrico que mide cambios en los niveles de radiación infrarroja emitida por los objetos a su alrededor a una distancia máxima de 6 m. (Sensor, 2012).

Figura 26. Sensor de Movimiento.



Fuente: <http://www.electronicaestudio.com/tienda/electronica/sensores/sensor-infrarrojo-pir-economico-detalles>

Tabla 2. Característica y aplicaciones del sensor de movimiento.

Aplicaciones	Características
<ul style="list-style-type: none">• Sistemas de Alarma.• Tableros publicitarios.• Iluminación activada por movimiento.• Aplicación en la robótica	<ul style="list-style-type: none">• Voltaje de alimentación: 5 DC• Nivel de salida: 0 - 3.3V.• Interface de 3 pins.• Compatible con ARDUINO.• Angulo de detección: 100°.• Rango detección: 6 m.• Método de activación : L evento irrepitable;H evento repetible• Temperatura ambiente soportada: 0°C a +50°C• Dimensiones placa: 32 X 24,3X 25,4mm.

Fuente: Autora.

2.11.2 Sensor Infrarrojo Sharp 2Y0A21. El sensor GP2D12 de sharp es un dispositivo de reflexión por infrarrojos que indica mediante una salida analógica la distancia medida. La tensión de salida varía de forma no lineal cuando se detecta un objeto en una distancia entre 10 y 80 cm.

La salida está disponible de forma continua y su valor es actualizado cada 32 ms. Normalmente se conecta esta salida a la entrada de un convertidor analógico digital el cual convierte la distancia en un número que puede ser usado por el microprocesador.

La salida también puede ser usada directamente en un circuito analógico. Hay que tener en cuenta que la salida no es lineal. El sensor utiliza solo una línea de salida para comunicarse con el procesador principal. (Super, 2011).

Figura 27. Sensor Infrarrojo Sharp.



Fuente: <http://meatronicadigital.com/esp/index/item/83/37/sensor-infrarrojo-de-proximidad-y-distancia-de-mediano-alcance>

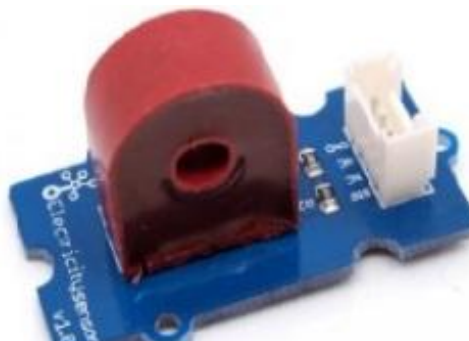
Tabla 3. Característica y aplicaciones del sensor infrarrojo Sharp

Aplicaciones	Características
En la Robótica.	<ul style="list-style-type: none"> Tensión de alimentación 4,5 a 5,5 V CC. Compatible con ARDUINO. Temperatura funcionamiento: -10 a 60°C. Rango de medición L 10 a 80cm. Consumo Medio: 35 mA.

Fuente: Autora.

2.11.3 Sensor de corriente análogo. Este sensor usa un transformador TA12-200, que puede cambiar una gran corriente alterna en una de pequeña amplitud. Se puede utilizar para probar corriente alterna de hasta 5 A. (Silicio, 2013)

Figura 28. Sensor de corriente análogo.



Fuente: <http://www.jola.ro/current/518-senzor-curent-groVe.html>

Tabla4. Característica y aplicaciones del sensor de corriente análogo

Aplicaciones	Características
<ul style="list-style-type: none"> Medición de la corriente alterna. Dispositivo de monitorización de estado. 	<ul style="list-style-type: none"> Entrada máxima 5 A. Alta precisión. 800Ω Pequeño tamaño. Compatible con ARDUINO. Voltaje de operación 5Vcd. Muy pequeño: 34 x 21 x 1.6mm. Interface de 3 pins.

Fuente: Autora.

2.11.4 Sensor fotoeléctrico. Un sensor fotoeléctrico es un dispositivo que se utiliza para detectar la distancia, la ausencia o presencia de un objeto mediante el uso de un transmisor de luz, a menudo de infrarrojos y un receptor fotoeléctrico. (NEXTIA, 2011)

Figura 29. Sensor fotoeléctrico.



Fuente: <http://www.dx.com/es/p/e18-d80nk-infrared-obstacle-avoidance-detection-photoelectric-sensor-for-Arduino-3-80cm-284985#.VH9D6LeUObw>

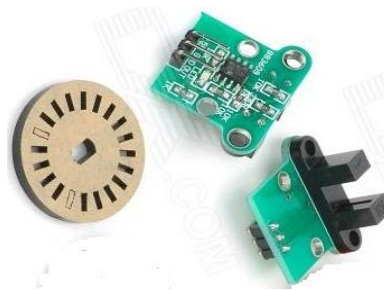
Tabla5. Característica y aplicaciones del sensor fotoeléctrico.

Aplicaciones	Características
<ul style="list-style-type: none"> Maquinaria textil La metalurgia El ferrocarril La industria de guerra, etc. 	<ul style="list-style-type: none"> Voltaje: 5VDC. corriente: 100mA. gama de medición: 3 a 80cm. el potenciómetro de ajuste trasero y el indicador de salida. definiciones del perno: el poder rojo, negro es Gnd o tierra, la salida amarilla de la señal.

Fuente: Autora.

2.11.5 Sensor fotoeléctrico Encoders. Este sensor se ha establecido como eficaces para tareas de posicionamiento de extrema precisión. Transforman movimientos de rotación en señales digitales. Los encoders funcionan sin desgaste mediante detección fotoeléctrica o magnética. Para ello disponen de un disco graduado que está sujeto firmemente en el eje o de un soporte magnético móvil. (Cyber, 2012)

Figura 30. Sensor fotoeléctrico Encoders.



Fuente: http://www.miniinbox.com/es/doble-Velocidad-hc-020k-medicion-modulo-w-fotoelectrico-encoders-negro-Verde-2-pcs_p647629.html#

Tabla6. Característica y aplicaciones del sensor Encoders.

Aplicaciones	Características
Móviles	<ul style="list-style-type: none"> • Voltaje de funcionamiento: 4.5 a 5.5V • Frecuencia de medición: 100 kHz; • Diámetro del disco: 24 mm; • Resolución encoder: 20 líneas; • Configuración del sensor de medición de velocidad: medida de la línea 1 la velocidad del motor.

Fuente: Autora.

2.11.6 Sensor LM35. El LM35 Es un sensor de temperatura integrado de precisión, cuya tensión de salida es linealmente proporcional a temperatura en °C (grados centígrados). El LM35 por lo tanto tiene una ventaja sobre los sensores de temperatura lineal calibrada en grados Kelvin. (Electrónica, 2012).

Figura 31. Sensor LM35



Fuente: <http://potentiallabs.com/cart/lm35-temp-sensor>

Tabla 7. Característica y aplicaciones del sensor LM 35.

Aplicaciones	Características
<ul style="list-style-type: none"> • Compensar un dispositivo de medida sensible a la temperatura ambiente. • Refrigerar partes delicadas del robot • Temperaturas en el transcurso de un trayecto de exploración 	<ul style="list-style-type: none"> • Calibrado directamente en grados Celsius (Centígrados) • Factor de escala lineal de +10 mV / °C • 0,5°C de precisión a +25 °C • Rango de trabajo: -55 °C a +150 °C • Funciona con alimentaciones entre 4V y 30V • Menos de 60 µA de consumo • Bajo auto-calentamiento (0,08 °C en aire estático)

Fuente: Autora.

2.12 Fundamentos de Mantenimiento.

Innumerables definiciones se le ha dado al mantenimiento con el pasar de los años, por lo que en la actualidad ejecutar el mantenimiento no implica el realizar una reparación al equipo averiado, sino mantenerlo operando bajo los niveles establecidos por los fabricantes, dentro de los índices de producción de una organización, la finalidad del mantenimiento es la de conservar y mantener operable un equipo en una condición particular, restablecerlo a las condiciones iniciales, logrando la máxima productividad siendo éstos; eficientes y capaces de cumplir con la función por las que fueron diseñados.

2.12.1 Objetivos del mantenimiento. Actualmente el mantenimiento por estar inmerso en todas las actividades empresariales, juega un papel cada vez más importante que abarca más responsabilidades y es así que sus objetivos van enmarcados a:

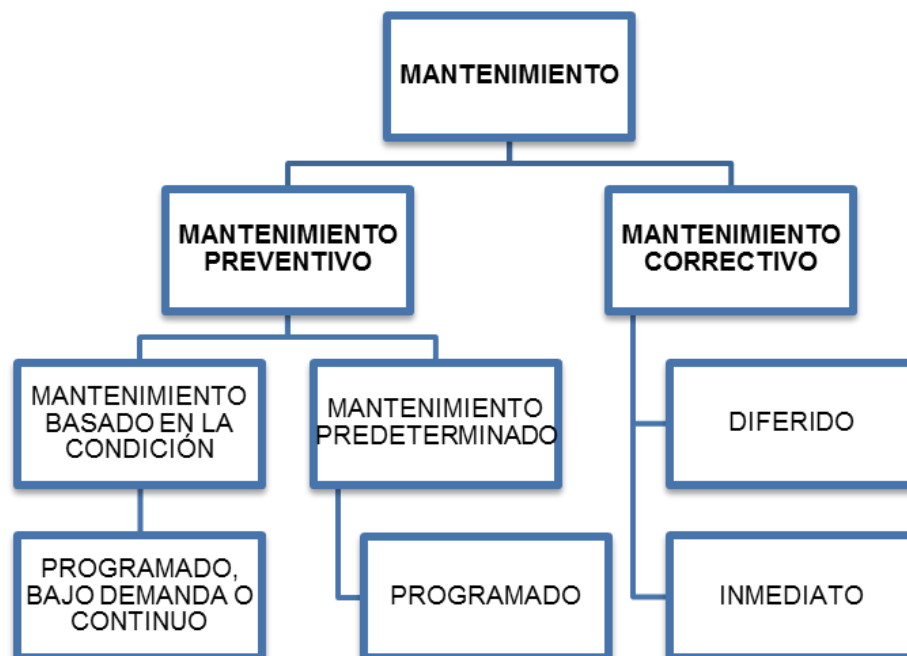
- La disponibilidad, mismo que determina el tiempo bruto de productividad.
- El costo, ya que los paros imprevistos presentan cuantiosas pérdidas.

- La seguridad, al disminuir el riesgo en el manejo y operación de los equipos.
- La integridad ambiental, al producir sin contaminar debido al buen estado en que se encuentran los equipos.
- Eficiencia energética, al proporcionar el funcionamiento eficiente de los equipos.
- La calidad de los productos, al no producir defectuosos.
- Servicios al cliente, por las entregas a tiempo.

2.12.2 Tipos de mantenimiento. Los mantenimientos mencionados a continuación tienen como objetivo preservar un artículo o restaurarlo a un estado en el cual pueda llevar a cabo alguna función requerida. Estos procedimientos incluyen la combinación de acciones técnicas y administrativas correspondientes.

A continuación se muestra una clasificación del mantenimiento.

Figura 32. Tipos de mantenimiento



Fuente: Norma AENOR UNE-EN 13306 (Anexo A p.20)

2.12.3 Mantenimiento Productivo Total (TPM). El TPM es una serie de actividades que ayudan a mejorar la competitividad de una empresa, manteniendo a los equipos en un alto índice de efectividad al operar, en esta estrategia la producción con mantenimiento van a la par lo que quiere decir es que tienen la misma importancia en una industria, logrando con ello el producir productos de calidad a menores costos y en el momento necesario.

El TPM tiene como objetivos el lograr:

- Cero accidentes.
- Cero defectos.
- Cero averías.

2.12.4 Manuales de mantenimiento. Un manual de mantenimiento describe las normas, la organización y los procedimientos que se utilizan en una empresa para efectuar la función de mantenimiento. Dicho manual eleva el papel del mantenimiento a un lugar muy importante de la organización, cuando los procesos se encuentran ordenados y son llevados a cabo de una manera satisfactoria.

2.12.4.1 objetivos del manual de mantenimiento. Como objetivo primordial se puede decir que la función de un manual de mantenimiento es la de proporcionar a la unidad de mantenimiento un sistema de procesos administrativos.

Mediante etapas de planeación, organización, ejecución, control e inspección, que contribuyan como un apoyo en las actividades de mantenimiento de las instalaciones y equipos de la empresa.

Mientras que como objetivos específicos se puede nombrar los siguientes:

- Controlar las actividades de mantenimiento realizadas a instalaciones y equipos.
- Facilitar las actividades del mantenimiento predictivo, preventivo y correctivo, ordenando cada uno de los procesos dentro de la unidad de mantenimiento.

- Establecer normas de seguridad relacionadas con el trabajo de mantenimiento.

2.12.4.2 Tipos de manuales de mantenimiento

- **Manual de mantenimiento predictivo:** Contempla las revisiones periódicas para detectar cualquier condición que pudiera impedir el uso apropiado y seguro del dispositivo y poder corregirla, manteniendo de ésta manera cualquier instalación, herramienta o equipo en óptimas condiciones de uso.
- **Manual de mantenimiento preventivo:** Contempla los ajustes, cambios modificaciones, limpieza y reparaciones (generalmente sencillos) necesarios para mantener cualquier instalación, herramienta o equipo en condiciones seguras de uso, con el fin de evitar posibles daños al operador o al equipo mismo.
- **Manual de mantenimiento correctivo:** Contempla las reparaciones, cambios o modificaciones de cualquier herramienta, maquinaria o equipo cuando se ha detectado alguna falla o posible falla que pudiera poner en riesgo el funcionamiento seguro de la instalación, herramienta o equipo y de la persona que lo utiliza. (ALBORNOZ SALAZAR, 2010)

CAPÍTULO III

3. DISEÑO, PROGRAMACIÓN E IMPLEMENTACIÓN: DEL MÓDULO DE CONTROL DE INGRESO DE USUARIOS AL LABORATORIO EN TIEMPO REAL Y TABLERO EQUIPADO CON VARIOS SENSORES, UTILIZANDO SOFTWARE, HARDWARE ARDUINO Y LabVIEW.

3.1 Análisis del software y hardware Arduino

Análisis del porqué el uso del software y hardware Arduino. El uso del software y hardware Arduino se decidió por los siguientes aspectos:

3.1.1 Análisis del Software Arduino. Es una plataforma electrónica abierta que sirve para la creación de prototipos fáciles de usar, basados en software y hardware flexible.

El uso del software Arduino se decidió por los siguientes aspectos:

- Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectarlo a un ordenador.
- Es fácil de usar y manejar el software a los usuarios avanzados.
- El software Arduino está publicado bajo una licencia libre.
- Para programar en Arduino se utiliza un lenguaje propio es decir creado específicamente para esta plataforma.
- En el software Arduino utiliza comandos AT y librerías para realizar el control.

3.1.2 Análisis del hardware Arduino. Es una placa electrónica que se la encuentra dependiendo de las necesidades del usuario como se muestra:

- Posee una diversidad de tarjetas como Arduino: uno, mega ADK etc; shields son placas que complementan a la tarjeta Arduino como GSM, etc.

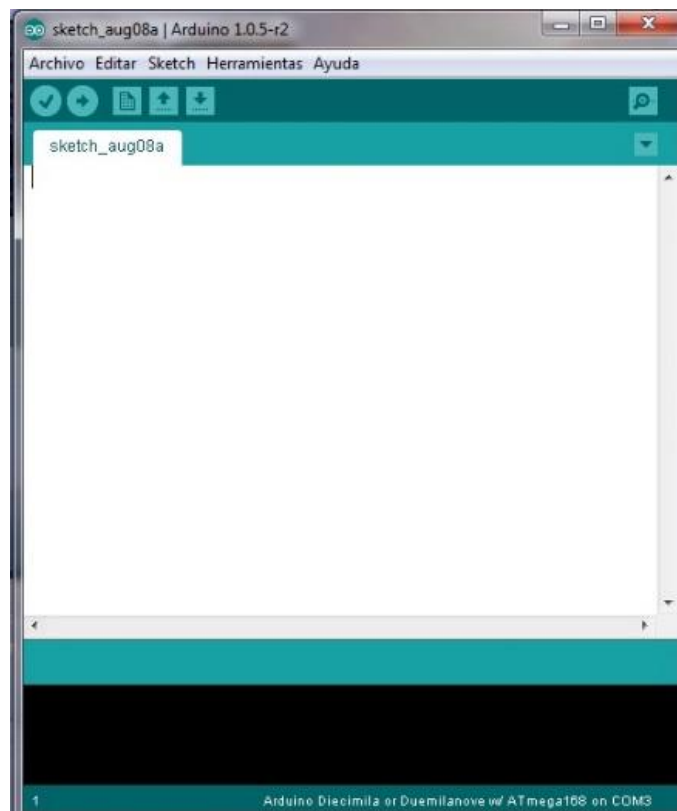
- Son tarjetas económicas asequibles a cualquier usuario.
- Presenta facilidad para adoptar shields.
- Tiene diversidad de sensores.

3.2 Aplicación con software Arduino

La Programación en el software Arduino se realiza con un lenguaje propio creado por esta plataforma, este lenguaje está basado en processing que es un lenguaje de alto nivel, para poder iniciar la programación se debe instalar el IDE (Entorno Interactivo de desarrollo) de Arduino. La ventana del IDE de Arduino consta de una barra: título, menús, estándar; zona de trabajo, área: de mensaje y de visualización de errores.

Pantalla principal de programación o IDE.- Aquí se detallará como se realizó la programación en Arduino, con la cual se logra el control del módulo, y posterior envío de notificaciones del ingreso de usuarios al laboratorio de Electrónica de la Facultad de Mecánica.

Figura 33.- Pantalla principal o IDE

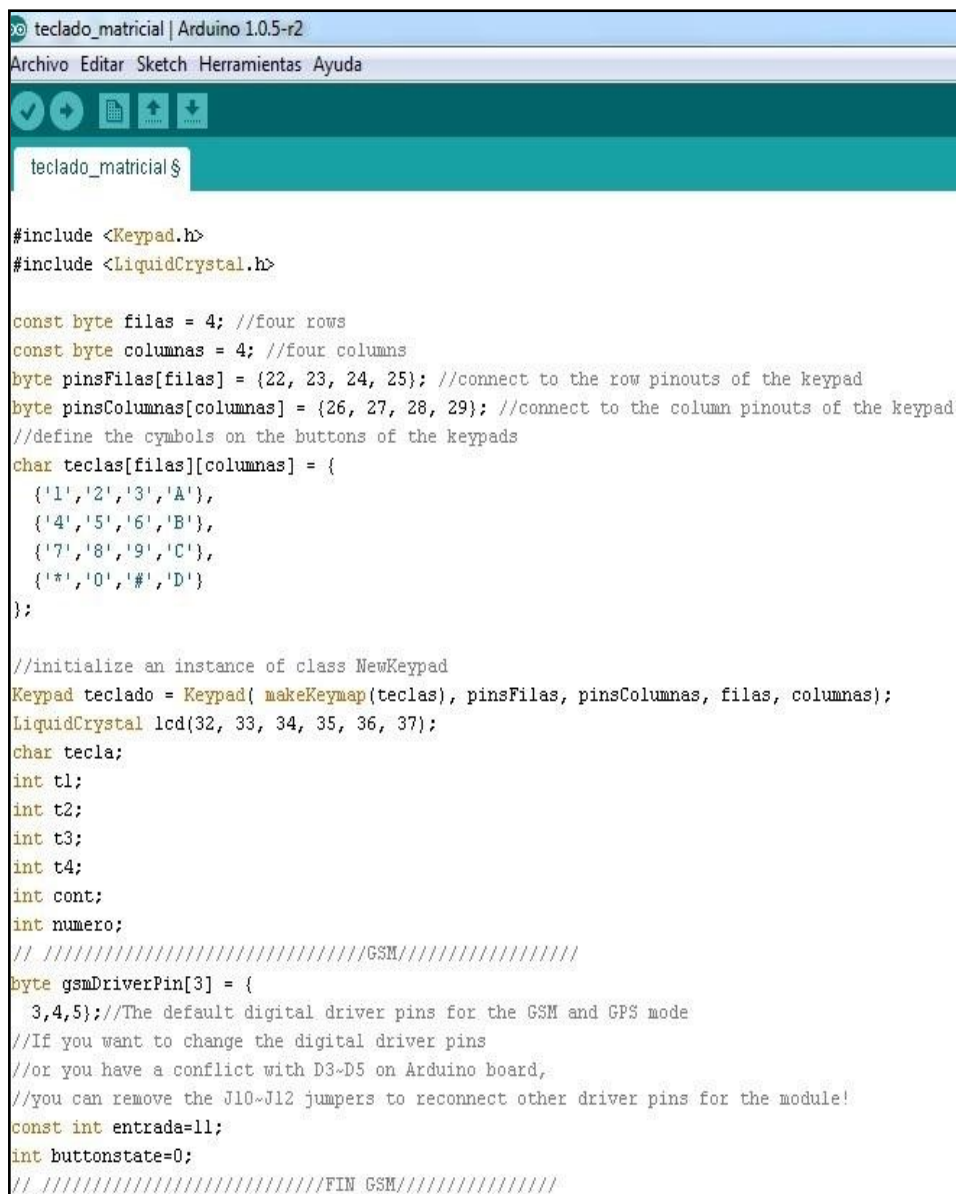


Fuente: Autora.

3.2.1 La programación que va a controlar el módulo se encuentra dividida en tres partes:

3.2.1.1 Primera parte. Aquí se enlaza las librerías que existen en el teclado y en la pantalla de cristal líquido al programa. A continuación definimos una serie de constantes que utilizamos para indicar el número de filas y columnas del teclado, luego indicamos los pines que van a ser conectados a cada fila y columna del Arduino Mega ADK, luego inscribimos la denominación para que se pueda conectar a la librería del teclado a la secuencia de programación. El paso siguiente a realizar en la programación es indicar que pines de la pantalla de cristal líquido o LCD van a ir conectados al Arduino.

Figura 34. Programación primer parte



```
teclado_matricial | Arduino 1.0.5-r2
Archivo Editar Sketch Herramientas Ayuda

teclado_matricial$

#include <Keypad.h>
#include <LiquidCrystal.h>

const byte filas = 4; //four rows
const byte columnas = 4; //four columns
byte pinsFilas[filas] = {22, 23, 24, 25}; //connect to the row pinouts of the keypad
byte pinsColumnas[columnas] = {26, 27, 28, 29}; //connect to the column pinouts of the keypad
//define the cymbols on the buttons of the keypads
char teclas[filas][columnas] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

//initialize an instance of class NewKeypad
Keypad teclado = Keypad( makeKeymap(teclas), pinsFilas, pinsColumnas, filas, columnas);
LiquidCrystal lcd(32, 33, 34, 35, 36, 37);
char tecla;
int t1;
int t2;
int t3;
int t4;
int cont;
int numero;

// //////////////////////////////////GSM////////////////////////////////////
byte gsmDriverPin[3] = {
  3,4,5}; //The default digital driver pins for the GSM and GPS mode
//If you want to change the digital driver pins
//or you have a conflict with D3~D5 on Arduino board,
//you can remove the J10~J12 jumpers to reconnect other driver pins for the module!
const int entrada=11;
int buttonstate=0;
// //////////////////////////////////FIN GSM////////////////////////////////////
```

Fuente: Autora.

3.2.1.2 Segunda parte. Determinar la ubicación del cursor en el LCD para indicar las letras que van en la primera fila y su respectiva declaración del estado de los pines.

Figura 35. Programación segunda parte

```
void setup() {  
  lcd.begin(16,2);  
  lcd.setCursor(0,0);  
  lcd.print("INGRESE # 0 COD.");  
  Serial.begin(9600);  
  cont=0;  
  t1=0;  
  t2=0;  
  t3=0;  
  t4=0;  
  //////////////////////////////////GSM////////////////////////////////////  
  pinMode (entrada, INPUT);  
  //Init the driver pins for GSM function  
  for(int i = 0 ; i < 3; i++){  
    pinMode(gsmDriverPin[i],OUTPUT);  
  }  
  digitalWrite(5,HIGH); //Output GSM Timing  
  delay(1500);  
  digitalWrite(5,LOW);  
  digitalWrite(3,LOW); //Enable the GSM mode  
  digitalWrite(4,HIGH); //Disable the GPS mode  
  delay(2000);  
  Serial.begin(9600); //set the baud rate  
  delay(5000); //call ready  
  delay(5000);  
  delay(5000);  
  //////////////////////////////////FIN GSM////////////////////////////////////  
}
```

Fuente: Autora.

3.2.1.3 Tercera parte: En el programa se indica el número de código que tiene el usuario al ingresar al laboratorio. En la pantalla se muestra el código del usuario en forma de un carácter.

Figura 36. Código de usuario

```
lcd.print("*");  
if (numero==8) {
```

Fuente: Autora.

- Si es correcto el código, luego se graba el número del usuario.

Figura 37. Número de usuario

```
Serial.println("AT+CMGS=\"0968967410\"); //Change the receiver phone number
```

Fuente: Autora.

- A continuación se programa el mensaje que se desea enviar al teléfono del encargado utilizando los comandos AT y finalmente en la pantalla LCD muestra un texto de confirmación que dice el mensaje ha sido enviado.

Figura 38. Mensaje de confirmación

```
Serial.print("Ingreso al Laboratorio: Personal Encargado"); //the message you want to send
```

Fuente: Autora.

- Luego de haber realizado los pasos anteriores el paso tres se muestra en la siguiente figura.

Figura 39. Programación de la tercera parte


```

void loop(){

  tecla= teclado.getKey();
  if (tecla != NO_KEY){
    numero= tecla-48;
    Serial.print(tecla);
    Serial.print(numero);
    lcd.setCursor(7,1);
    lcd.print("#");
    if (numero==8){
      Serial.println("AT"); //Send AT command
      delay(2000);
      Serial.println("AT");
      delay(2000);
      Serial.println("AT+CMGF=1");
      delay(1000);
      Serial.println("AT+CMGS=\"0968967410\"");//Change the receiver phone number
      delay(1000);
      Serial.print("Ingreso al Laboratorio: Personal Encargado");//the message you want to send
      delay(1000);
      Serial.write(26);
      lcd.setCursor(0,1);
      lcd.print("Mensaje Enviado ");
      delay(2000);
      lcd.setCursor(0,1);
      lcd.print("          ");
    }
  }
}

```

Fuente: Autora.

3.3 Especificaciones técnicas de las tarjetas ArduinoShield y Arduino MEGA

3.3.1 Módulo V3.0. La GPS/GPRS/GSM shield está basada en un módulo cuatribanda SIM 908 que funciona en la frecuencias 850/900/1800/1900 MHz. También incorpora un receptor GPS para recuperar datos de posicionamiento. La shield permite comunicarse con el mundo exterior por la red móvil, todo al mismo tiempo.

Tabla 8. Especificaciones técnicas de la tarjeta Shield

Alimentación	6-12 V
Consumo en reposo	100 mA / 7 V
Quad-Banda	850/900/1800/1900 MHz
Soporta fase GSM	2/2+
Clase 4	2 W a 850/900 MHz
Clase 1	1 W a 1800/1900 MHz
Control con comandos AT	GSM 07.07, 07.05 y SIM COM
Receptor GPS	Incorporado
Antena SMD integrada	GSM y GPS
Conexión para teclado	4 x 4
Interruptor USB	Arduino

Indicadores LED para alimentación	Estado: Red y módulo
-----------------------------------	----------------------

Fuente: Autora.

3.3.2 Arduino MEGA ADK. Es una placa electrónica basada en el microprocesador Atmega2560 fabricada también por ATMEL, es conocido por muchos por ser el cerebro de las impresoras 3D, gracias a sus 54 pines de entrada o salida permite conectar un gran número de componentes a la placa.

Tabla 9. Especificaciones técnicas de Arduino Mega

Micro controladores	Atmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Digital pines I / O	54 (de las cuales 15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente DC por Pin I / O	40 mA
Corriente DC de 3.3V Pin	50 mA
Memoria Flash	256 KB de los cuales 8 KB utilizado por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
USB Host chip	MAX3421E

Fuente: Autora.

3.4 Pruebas de funcionamiento del programa que controla el módulo de control de ingresos de usuarios.

Luego de haber desarrollado el programa, se realizaron diversas pruebas al mismo, para verificar el correcto funcionamiento, las pruebas se realizaron con ayuda de un protoboard donde todos los elementos fueron conectados entre sí como lo podemos ver en la figura.

Figura 40. Pruebas realizadas en el protoboard.



Fuente: Autora.

Después de haber realizado las pruebas del programa, se puede concluir que cumple satisfactoriamente con las necesidades requeridas ya que si envía el mensaje de notificación de ingreso de usuarios a los números programados, realizando el respectivo control deseado.

3.5 Listado de los equipos, herramientas y materiales.

Para la implementación en el módulo de ingreso de usuarios se requirió de los siguientes equipos, herramientas y materiales, a continuación se muestra la tabla.

Tabla 10. Listado de los equipos, herramientas y materiales.

Cantidad (unidades)	Descripción
2	Caja de paso rápido Lock 200 X 200 X 90 mm.
2	Tarjeta MEGA ADK.
2	Módulo V 3.0.
2	Pantalla LCD
2	Teclado matricial de 4 X 4
2	Adaptadores inteligentes de voltaje variable.
40	Cable mixto para conexiones electrónicas.
2	Cable de datos USB 28 awg.

8	Pernos
2	Base de acrílico.
1	Rollo de cinta doble fas.
1	Una canaleta de 10 X 10 mm
1	Alicate.
1	Pinzas puntas planas.
1	Pinzas puntas redondas.
1	Destornillador de estrella.
1	Destornillador plano
2	Juego de piedras pulidoras.
1	Estilete.
1	Cinta aislante

Fuente: Autora.

3.6 Implementación del módulo de control de ingreso de usuarios.

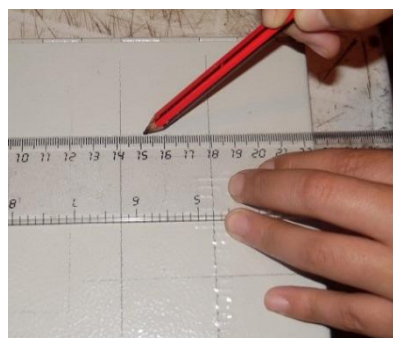
A continuación se detalla el procedimiento con el cual se modificó parte de la infraestructura física inmersa en el proyecto.

El proyecto trata del control de ingreso de usuarios al laboratorio de Electrónica de la Facultad de Mecánica, es necesario la instalación de un módulo de control de ingresos con tarjetas Arduino, ya que se debe cuidar y prevenir de cualquier daño o pérdida de los activos que se encuentran dentro del laboratorio, el control se lo va a realizar mediante la red GSM y el mensaje de notificación se recibirá en cualquier parte del territorio ecuatoriano.

3.6.1 Montaje de los diferentes componentes en el módulo.

- Para el montaje de los diferentes elementos se designó los lugares que ocuparan mediante líneas lim itantes, esta ubicación se encuentra en diferentes partes del según los requerimientos necesarios. Como se muestra en la figura.

Figura41.Ubicación de la pantalla y teclado



Fuente: Autora.

Luego de haber designado los respectivos lugares que ocuparan los elementos en el módulo, se realizó los agujeros de acuerdo a las dimensiones de los elementos.

- Realización de las perforaciones para la pantalla y el teclado.- la ubicación de estos se encuentra en la parte frontal del módulo.

Figura 42. Perforaciones para la pantalla y teclado.



Fuente: Autora.

- Realización de las perforaciones para el interruptor y potenciómetro.- según la facilidad de manipulación el interruptor ON/OFF y el potenciómetro se encuentran en el costado derecho del módulo como se muestra en la figura.

Figura 43. Perforaciones para el interruptor y potenciómetro.



Fuente: Autora.

- Orificio para el cable que ira conectado a un tomacorriente y puesta de protección del mismo.

Figura 44. Orificio para el cable de alimentación



Fuente: Autora.

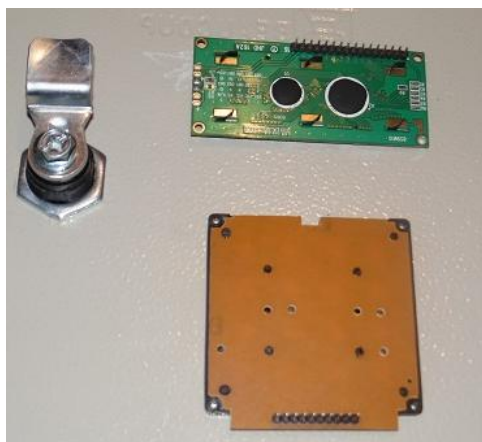
Figura 45. Protección del cable



Fuente: Autora.

- Luego de haber realizado los agujeros, se montó la pantalla LCD, el teclado, interruptor y potenciómetro en el módulo.

Figura 46. Montaje de la pantalla LCD y teclado, lado posterior



Fuente: Autora.

Figura 47. Montaje de la pantalla LCD y teclado parte frontal



Fuente: Autora.

- Montaje del interruptor ON/OFF y del potenciómetro.- por motivos de seguridad al módulo de conecto un interruptor de encendido y apagado, la función del potenciómetro es regular el brillo de la pantalla LCD.

Figura 48.Montaje del interruptor ON/OFF y del potenciómetro.



Fuente: Autora.

Para obtener el voltaje requerido con el que las tarjetas Arduino funcionen de una manera adecuada se conectó una extensión a un regulador de voltaje en la parte interna del módulo, tomando en cuenta el espacio que se dispone, se la ubico en la parte superior como se muestra en la figura.

Figura 49. Montaje del regulador de voltaje



Fuente: Autora.

3.6.2 Montaje de la tarjeta Arduino mega ADK y shield V3.0. Para el montaje de las tarjetas en el módulo se siguieron los siguientes pasos:

- Se Insertó la tarjeta shield V3.0 en la tarjeta Arduino mega ADK.- La tarjeta shield V3.0 es un complemento cuya función es la de comunicarse con el mundo exterior mientras que la tarjeta mega ADK contiene el programa que va a ser utilizado.

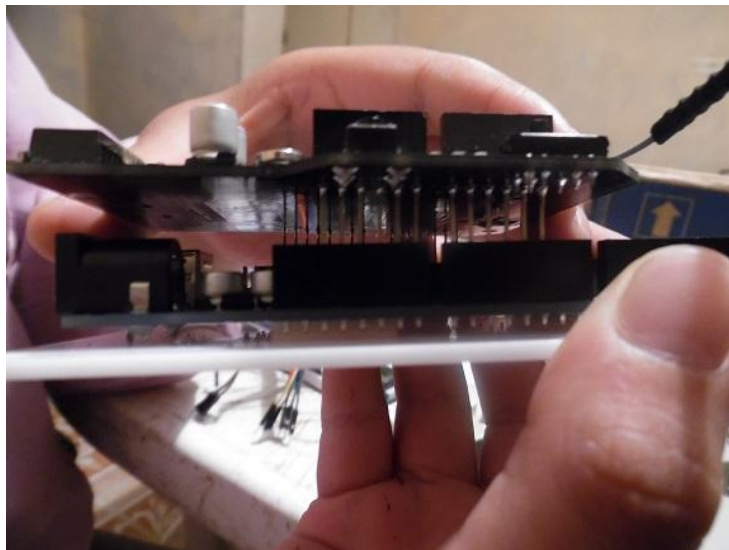
Figura 50.- tarjeta shield insertada en la tarjeta mega ADK



Fuente: Autora.

- Las tarjetas Arduino y shield se colocaron en una base de acrílico para protegerles, de que las pistas no vayan a tocar ninguna parte metálica, evitando que se quemen las tarjetas por cortocircuito.

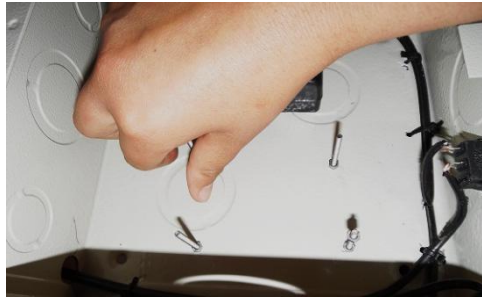
Figura 51. Unión de las tarjetas en la base



Fuente: Autora.

- En la base de acrílico se realizaron agujeros, que servirán, como guías para introducir pernos los cuales soportaran las tarjetas.

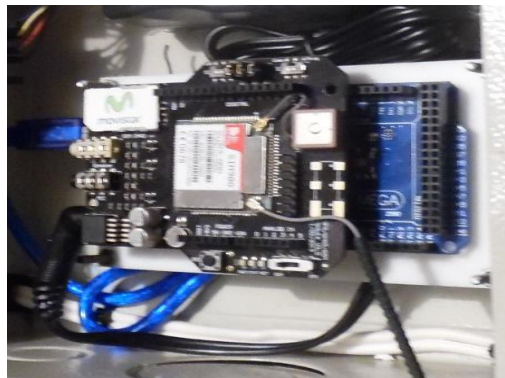
Figura 52. Montaje pernos de soporte.



Fuente: Autora.

- Finalmente se montó las tarjetas Arduino en conjunto con la base en los pernos de soporte, conectando previamente el conductor del regulador de voltaje y el cable de datos en sus respectivos lugares.

Figura 53. Montaje de las tarjetas en los pernos de soporte

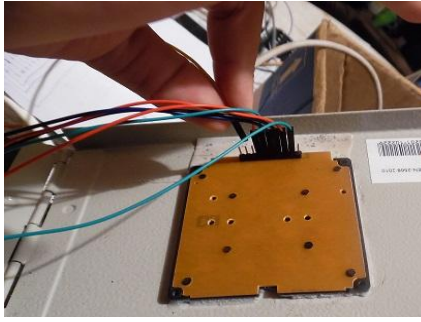


Fuente: Autora.

3.6.3 Conexiones de los elementos del módulo a las tarjetas Arduino.

- Luego de haber realizado el montaje todos los elementos en el módulo se realizó las respectivas conexiones a las tarjetas Arduino.

Figura 54. Conexión del teclado



Fuente: Autora.

Figura 55. Conexión de la pantalla LCD



Fuente: Autora.

Figura 56. Conexión del potenciómetro



Fuente: Autora.

- Para realizar la conexión necesitamos de una tarjeta electrónica que nos sirva como puente ya que los cables de conexión son pequeños. Los cables se conectan a la tarjeta electrónica y de la misma se conecta al Arduino.

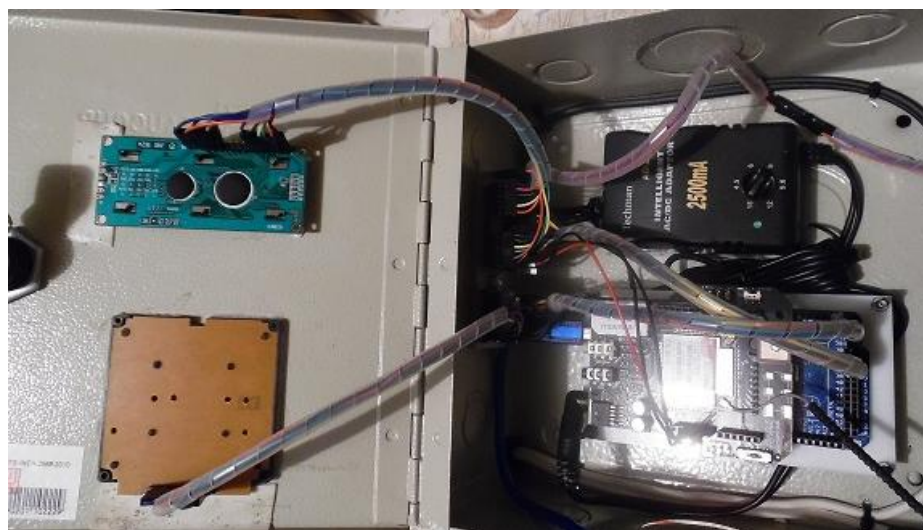
Figura 57. Tarjeta electrónica



Fuente: Autora.

- Por último al realizar todas las conexiones y comprobar el buen funcionamiento del módulo organizamos los cables conectados con la ayuda de organizadores de plástico finalizando así la construcción del módulo de control de ingresos de usuarios. Como se ve en la figura.

Figura 58.- Módulo de control de ingresos



Fuente: Autora.

3.6.4 *Instalación del módulo en el laboratorio.* La instalación física del módulo de control de ingreso de usuarios al laboratorio de Electrónica de la Facultad de Mecánica se realizó como lo muestran las siguientes figuras

Figura 59.- Realización de los agujeros para el empujado del módulo.



Fuente: Autora.

Figura 60.- Empotrado del módulo.



Fuente: Autora.

Figura 61.- Módulo en funcionamiento

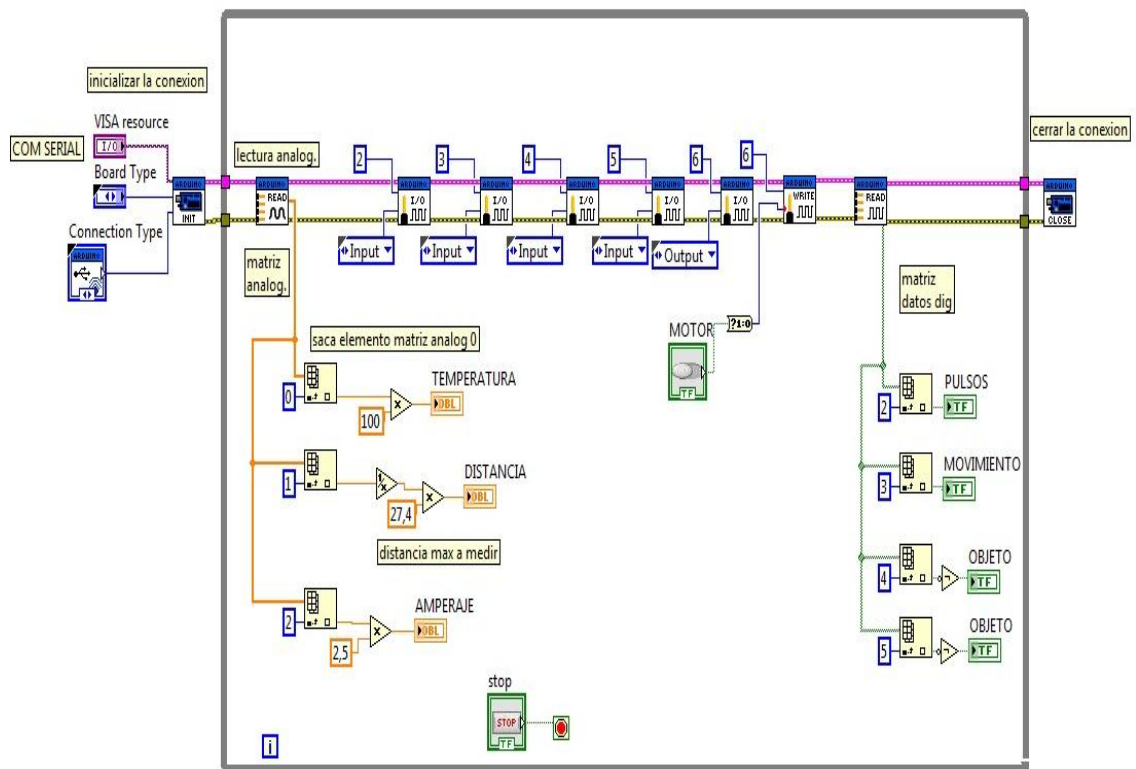


Fuente: Autora.

3.7 Tablero equipado con varios sensores, controlados con LabVIEW Y Arduino.

3.7.1 Aplicación con software LabVIEW .Para realizar la programación en software labVIEW : Se efectuó una interface entre labVIEW y Arduino. La programación se encuentra dentro de una estructura de repetición llamada estructura Whipe Look, como se muestra en la figura.

Figura 62. Diagrama de programación.



Fuente: Autora.

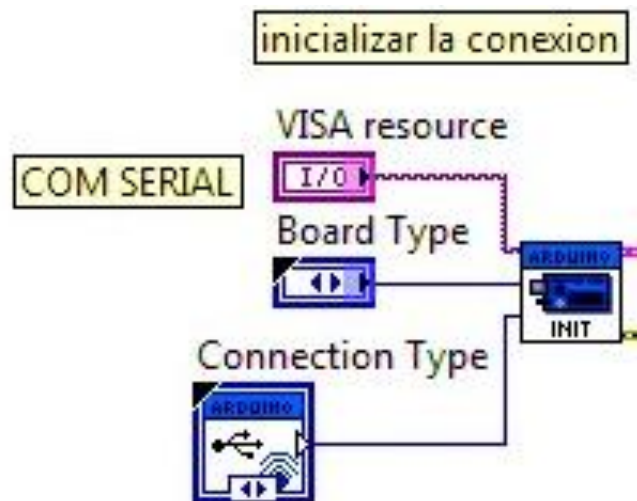
El desarrollo del programa que controla a los distintos tipos de sensores se detalla de una manera comprensiva a continuación.

La programación se lo realiza con un bloque de inicialización de conexión en este bloque tenemos:

- **VISA resource.** Especifica el puerto com, esto quiere decir a que puerto USB va a ir conectado el cable de datos.
- **Boardtype.** Indica el tipo de tarjeta que se está utilizando, en este proyecto la tarjeta a utilizarse es el Arduino Uno.
- **Connectiontype.** Indica la conexión serial que maneja el Arduino, esto quiere decir si se sesta controlando mediante un puerto USB o mediante Bluetooth, en este caso se está controlando con un puerto USB.

Los tres comandos mencionados anteriormente se conectan a un *INIT* (comando que ejecuta la interfaz entre LabVIEW y Arduino). En este paso se inicializa la conexión.

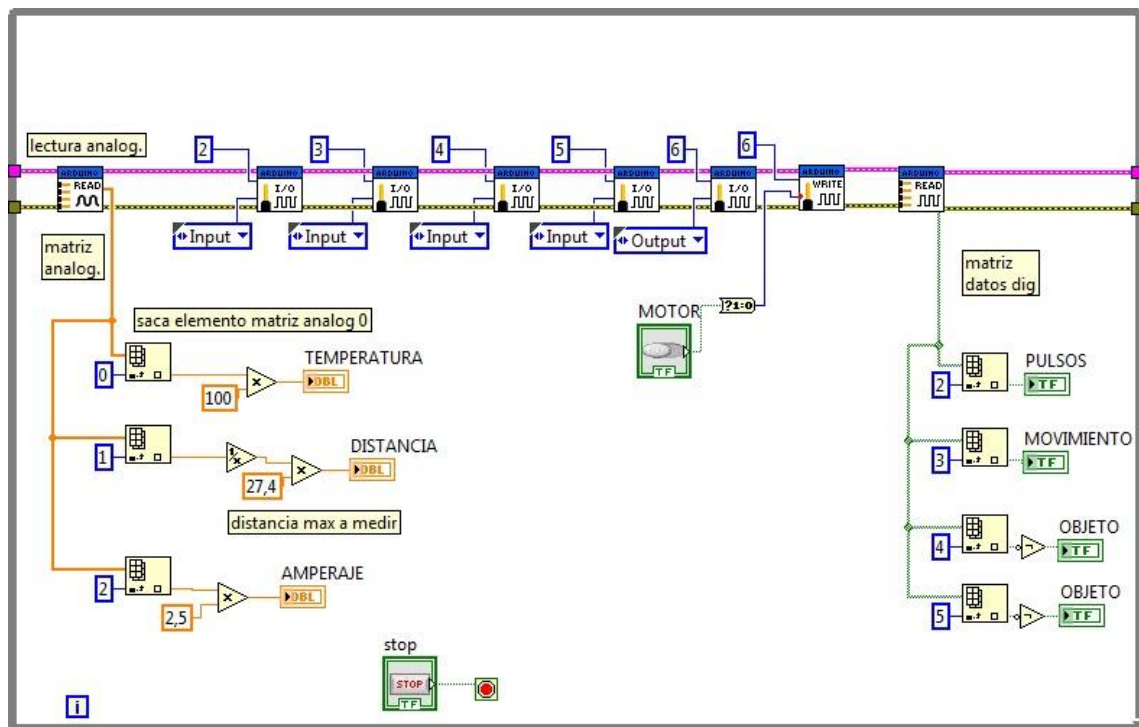
Figura 63. Bloque de Inicialización de conexión.



Fuente: Autora.

- A continuación en el WhileLoop (repite el programa hasta que el operario lo detenga o presione el botón de pare) dentro de esta estructura se encuentra:
- AnalogRead Port. Lee puertos analógicos de la matriz analógica y conecta a tres IndexArray, con su respectivo pin de conexión al Arduino y su respectivo factor de conversión los tres IndexArray conectados controlan los sensores analógicos los cuales son: sensor de temperatura, distancia y amperaje.
- Después de haber programado el AnalogRead Port, se une cuatro comandos denominados, Set Digital Pin mode que van uno a continuación de otro, estos comandos se encargan de leer las entradas de los sensores digitales los cuales son: sensor de pulsos, movimiento, distancia1 y distancia 2.
- A continuación del Set Digital Pin mode va programado un Digital read Port que se encarga de leer los valores digitales del Arduino.
- Luego se conecta a una matriz de cuatro IndexArray con sus pines de conexión ya preestablecidos para poder leer datos digitales en la pantalla frontal del LabVIEW.
- Dentro de esta estructura se indica el tiempo de ejecución de cada dato con una duración de 100ms tiempo para que siga la ejecución del programa cómo se indica en la siguiente figura.

Figura 64. Programación de lectura de datos analógicos y digitales.



Fuente: Autora.

- Para poder cerrar el programa se utilizó el comando de cerrado llamado CLOSE.

Figura 65. Comando close.

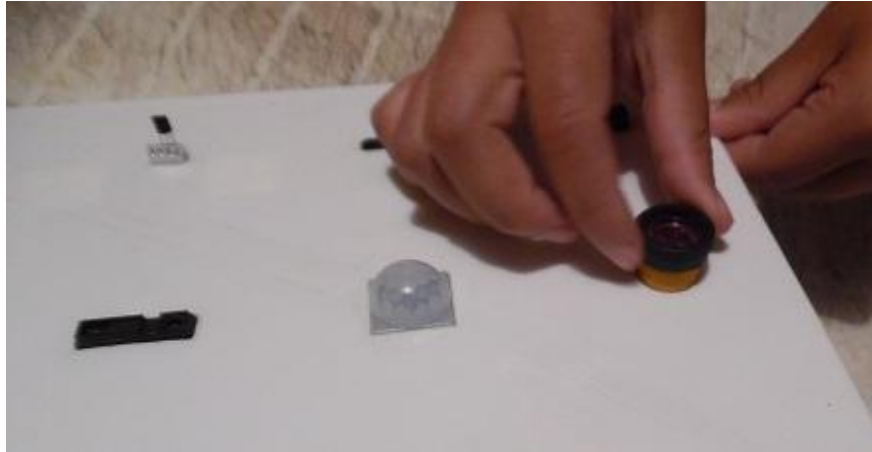


Fuente: Autora.

3.7.2 Implementación del tablero con sensores. La implementación de un tablero equipado con sensores de varios tipos y tarjetas Arduino con conexión a LabVIEW es de gran ayuda en el desarrollo de prácticas en las cátedras de Electrónica, Automatización y Mantenimiento. La estructura del tablero que aloja los sensores y tarjeta Arduino es de material acrílico, las características de este material son muy buenas aislando la corriente. A continuación se detalla los pasos para implementación:

3.7.2.1 Montaje de los sensores en el tablero. El tablero se lo realizo en un lugar que se especializa en la manufacturación con acrílico, es por eso que el montaje se lo realizo de la siguiente manera como nos muestra la figura.

Figura 66. Montaje de sensores



Fuentes: Autora.

3.7.2.2 Montaje de la tarjeta Arduino. La tarjeta Arduino que se utilizó en el control de sensores es el Arduino Uno.

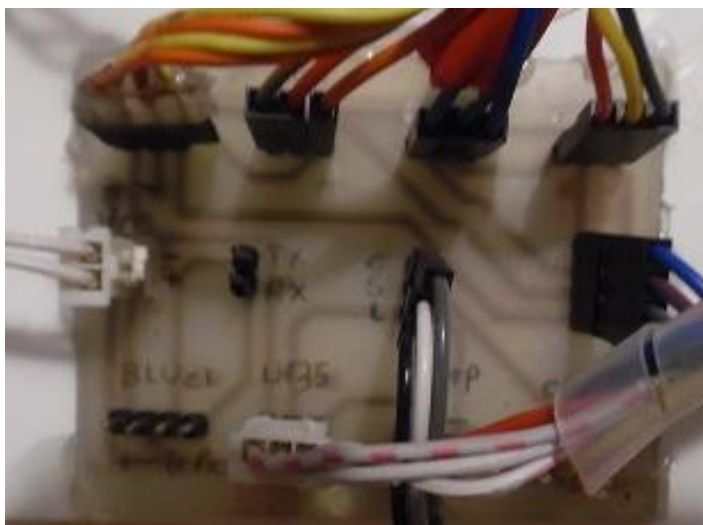
Figura 67. Montaje de la tarjeta Arduino



Fuentes: Autora

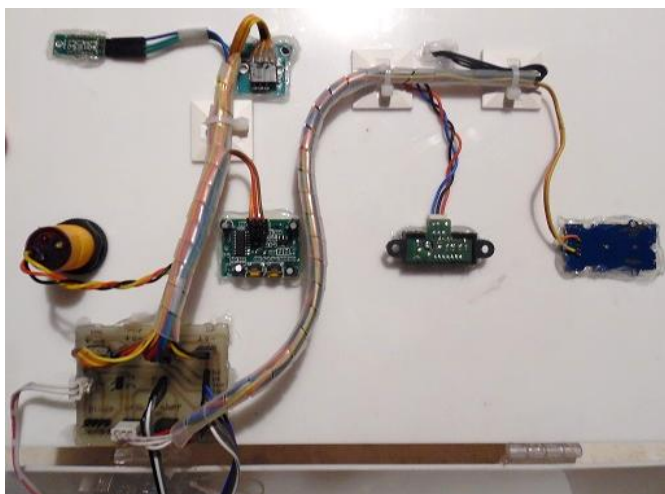
3.7.2.3 Adaptación de una tarjeta electrónica. El uso de esta tarjeta electrónica es porque los cables son muy cortos, esta tarjeta nos ayuda como un puente para la conexión posteriormente.

Figura 68. Tarjeta electrónica en el tablero de sensores



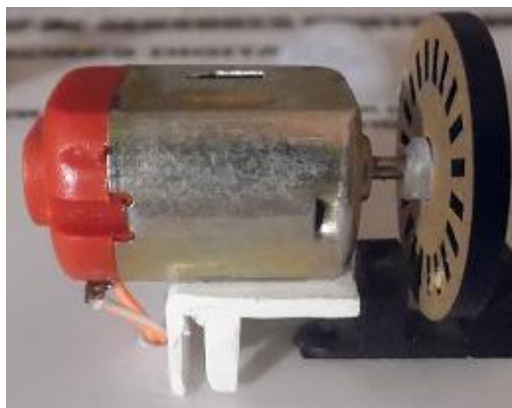
Fuente: Autora.

Figura 69. Conexión de los diferentes sensores



Fuente: Autora.

Figura 70. Montaje de un motor de 9 V cc



Fuente: Autora.

Figura 71. Montaje de la batería y potenciómetro



Fuente: Autora.

Figura 72. Módulo de sensores



Fuente: Autora.

CAPÍTULO IV

4. ELABORACIÓN DE MANUALES DE OPERACIÓN Y MANTENIMIENTO

4.1 Introducción

El manual de operación y mantenimiento en cualquier industria, taller o laboratorio tiene un alto grado de importancia, ya que en el manual de operación es una guía para poder realizar adecuadamente la ejecución del módulo y el manual de mantenimiento depende la producción, calidad de la producción, eficiencia, etc. El mantenimiento consiste en mantener una calidad óptima del producto/servicio que una máquina está ofreciendo, por tanto, hay que prevenir cualquier anomalía que se presente en la misma.

Tanto por el uso y por el tiempo un equipo está presto a sufrir una avería, falla o daño acortando su vida útil y su buen funcionamiento. Existen en forma general 2 tipos de mantenimiento, el mantenimiento preventivo y el correctivo; El mantenimiento correctivo es la reparación de la máquina ante alguna falla, el mantenimiento preventivo es la prevención de tal falla.

El manual de operación representa una pauta ya que nos indica la forma correcta como operar el módulo al personal que esté a cargo y el manual de mantenimiento tiene por objetivo que el personal que lo ocupe desarrolle destrezas y habilidades al realizar el mantenimiento del módulo instalado para el control y envío de notificaciones de usuarios al laboratorio de electrónica de la Facultad de Mecánica, ayudando a facilitar el uso correcto y el adecuado mantenimiento de los mismos.

4.2 Manual de operación de los módulos de control de ingreso de usuarios.

La elaboración de un manual de operación es necesario ya que muestra el proceso de cómo se debe operar los módulos de control y envío notificaciones de ingreso de usuarios con tarjetas Arduino al laboratorio de electrónica de la Facultad de Mecánica, desde su conexión al tomacorriente, hasta la visualización del funcionamiento en el respectivo módulo. (Ver anexo A)

4.3 Elaboración del manual de mantenimiento de los módulos de control

La elaboración del manual de mantenimiento es muy importante para mi proyecto, ya que se refiere a las medidas y acciones que se tomarán para mantener funcionando adecuadamente el módulo, sin que se emita errores con frecuencia.




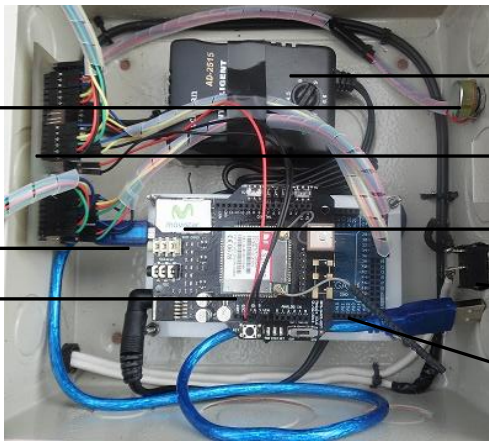
4.3.1 Fichas técnicas de los módulos adquiridos con sus respectivas partes. Las fichas técnicas son una valiosa ayuda, ya que en ellas se encuentra toda la información sobre los módulos como: marca, serie, código, procedencia, color, etc., a continuación en la tabla se muestra el modelo de la ficha técnica que se va a usar para el módulo implementados en este trabajo de grado.

Tabla 11. Ficha técnica de los módulos de control de ingreso.

 	MÓDULO DE CONTROL Y ENVÍO DE NOTIFICACIONES DE INGRESO DE USUARIOS	Ficha 1-1	
	DATOS TÉCNICOS – PARTES PRINCIPALES	Sección: Laboratorios de electrónica.	
Versión: 2010	FACULTAD DE MECÁNICA		Inventario:
		Manuales de Fabricante: Si	
FOTOGRAFÍA DEL EQUIPO			
	DATOS DE MÁQUINA		
	Marca	Modelo	Serie
	BEAUCOUP		
	Color	País de origen	Capacidad
	Gris	Ecuador	
DATOS DEL MÓDULO			
Marca			
Voltaje	9V		
DATOS IMPORTANTES (INTERIOR DEL MODULO)			
			
#	DENOMINACIÓN		
1	Pantalla LCD		
2	Teclado matricial 4X4		

Fuente: Fichas técnicas de la Facultad de Mecánica.

Tabla 12. Ficha técnica de los módulos de control de ingreso.

<div></div>		<div>MÓDULO DE CONTROL Y ENVÍO DE NOTIFICACIONES DE INGRESO DE USUARIOS</div>		<div>Ficha 1-1</div>	
				<div>Sección: Laboratorios de electrónica.</div>	
				<div>Inventario :</div>	
				<div>Manuales de Fabricante: Si</div>	
<div>Versión: 2010</div>		<div>FACULTAD DE MECÁNICA</div>			
<div>FOTOGRAFÍA DEL EQUIPO</div>		<div>DATOS DE MÁQUINA</div>			
<div></div>		<div>Marca</div>		<div>Modelo</div>	
		<div>BEAUCOUP</div>			
		<div>Color</div>		<div>País de origen</div>	
		<div>Gris</div>		<div>Ecuador</div>	
		<div>DATOS DEL MÓDULO</div>			
		<div>Marca</div>			
		<div>Voltaje</div>		<div>9V</div>	
<div>DATOS IMPORTANTES (INTERIOR DEL MÓDULO)</div>					
<div></div>					
<div>#</div>		<div>DENOMINACIÓN</div>			
<div>1</div>		<div>Regulador de voltaje</div>			
<div>2</div>		<div>Placa electrónica</div>			
<div>3</div>		<div>Sim (chip)</div>			
<div>4</div>		<div>Interruptor ON/OFF</div>			
<div>5</div>		<div>Arduino Mega</div>			
<div>6</div>		<div>Potenciometro B20K</div>			
<div>7</div>		<div>Cable de datos (USB)</div>			
<div>8</div>		<div>Shield V3.0</div>			



Fuente: Fichas técnicas de la Facultad de Mecánica.

4.3.2 Tareas de mantenimiento para los módulos implementados. La implementación del mantenimiento dentro de los laboratorios minimiza el riesgo de fallo y asegura la continua operación de los módulos, evitando su continua calibración

especialmente en equipos sensibles a las condiciones del entorno, a la incorrecta manipulación o a su inevitable desgaste.

4.3.3 Banco y ejecución de las tareas de mantenimiento de los módulos

Tabla 13. Banco de tareas para los módulos implementados en los laboratorios de Electrónica de la Facultad de Mecánica.

 	BANCO DE TAREAS DE MANTENIMIENTO PARA LOS MÓDULOS DE CONTROL IMPLEMENTADOS.
	TABLA DE TAREAS DE MANTENIMIENTO
Versión: 2010	LABORATORIO DE ELECTRÓNICA DE LA FACULTAD DE MECÁNICA

EQUIPO O MÁQUINA		
MODULO DE CONTROL DE INGRESO		
PARTES IMPORTANTES	TAREA DE MANTENIMIENTO	FRECUENCIA
Carcasa	Revisar y limpiar la carcasa	Diariamente
Circuito electrónico	Revisar y limpiar	Mensualmente
Placas electrónicas	Medir el voltaje	Mensualmente
Tarjetas Arduino	Revisar la conexión, voltaje	Mensualmente
Pantalla LCD	Cambiar	2 años
Teclado Matricial	Cambiar	2 años
Regulador de voltaje	Revisar y limpiar	Mensualmente
Potenciometro	Cambiar	Semestralmente

Fuente: Manuales de Mantenimiento de la Facultad de Mecánica.

A continuación la siguiente tabla muestra una guía para realizar el debido mantenimiento de los módulos instalados en el Laboratorio de Electrónica de la Facultad de Mecánica.

Para llevar a cabo estos propósitos se deben tener en cuenta los siguientes parámetros generales:

El punto más importante con respecto a la acción preventiva es la inspección de voltaje, revisar el voltaje de una manera periódica lograra que las tarjetas Arduino no se quemen o lleguen a desprogramarse dejando fuera de funcionamiento el módulo.

Tabla 14. Ejecución de las tareas de mantenimiento, medir el voltaje de las tarjetas Arduino

 	EJECUCIÓN DE LAS TAREAS DE MANTENIMIENTO		Diariamente	
			Mensualmente	X
	MODULO DE CONTROL		Trimestralmente	
			Semestralmente	
			Anual o más	
Versión: 2010	LABORATORIO DE ELECTRÓNICA DE LA FACULTAD DE MECÁNICA			

		Equipo	
		Apagado	X
		Encendido	
Tiempo de ejecución: 15 min			
MEDIR EL VOLTAJE DE LAS TARJETAS ARDUINO.			
Herramientas:	Materiales:	Equipos:	
		Multímetro	
Procedimiento:			
<ul style="list-style-type: none">No desconectar el adaptador de alimentación entre la tarjeta Arduino y el regulador de voltaje, con ayuda del multímetro comprobar que sea 9V.			
Observaciones:			
<ul style="list-style-type: none">Usar guantes para la ejecución de esta tarea			

Fuente: Ejecución de las tareas de mantenimiento de la Facultad de Mecánica.



- Condiciones ambientales.** Analizar el ambiente en el que se encuentra el equipo o módulo, ya sea en funcionamiento o en almacenamiento. Se recomienda evaluar temperatura, humedad, presencia de polvo y seguridad de la instalación.

Inspección externa del equipo. Examinar o reconocer atentamente el módulo, partes o accesorios que se encuentran a la vista, sin necesidad de quitar las partes, tornillo de sujeción, enchufe de corriente, etc.

- Limpieza de superficie externa usando una franela o paño húmedos.

Limpieza externa. Eliminar cualquier vestigio de suciedad, desechos, polvo, etc., en las partes externas que componen al módulo, mediante los métodos adecuados según corresponda.

Tabla 15. Ejecución de las tareas de mantenimiento, revisión y limpieza externa del módulo de control



 	EJECUCIÓN DE LAS TAREAS DE MANTENIMIENTO		Diariamente	
			Mensualmente	X
	MÓDULO DE CONTROL		Trimestralmente	
			Semestralmente	
			Anual o más	
Versión: 2010	LABORATORIO DE ELECTRÓNICA DE LA FACULTAD DE MECÁNICA			

		E q u i p o	
		Apagado	X
T i e m p o d e e j e c u c i ó n : 15 m i n		Encendido	
REVISIÓN Y LIMPIEZA EXTERNA DEL MÓDULO .			
Herramientas :	Materiales :	Equipos :	
Destornillador de estrella	Brocha, Fanela		
Procedimiento : <ul style="list-style-type: none">- Desconectar el cable de alimentación .- Revisar que los cables externos estén bien conectados .- Revisar la luminosidad de la pantalla LCD .- Limpiar la superficie externa de la carcasa con la ayuda de una fanela .- Verificar si el los tornillos de sujeción o empotramiento están flojos, ajustarlo si es necesario .			
Observaciones : <ul style="list-style-type: none">- Usar guantes para la ejecución de esta tarea			

Fuente: Ejecución de las tareas de mantenimiento de la Facultad de Mecánica.

Inspección interna del equipo. Examinar o reconocer atentamente las partes internas del equipo y sus componentes, para detectar signos de corrosión, o piezas faltantes; o cualquier signo que obligue a sustituir las partes afectadas o a tomar alguna acción pertinente al mantenimiento preventivo o correctivo.

Tabla 16. Ejecución de las tareas de mantenimiento revisión y limpieza interna del módulo de control

 	EJECUCIÓN DE LAS TAREAS DE MANTENIMIENTO		Diariamente	
			Mensualmente	X
	MÓDULO DE CONTROL		Trimestralmente	
			Semestralmente	
	Versión: 2010		Anual o más	
		LABORATORIO DE ELECTRÓNICA DE LA FACULTAD DE MECÁNICA		

		Equipo	
		Apagado	X
Tiempo de ejecución: 15 min		Encendido	
REVISIÓN Y LIMPIEZA INTERNA DEL MÓDULO.			
Herramientas:	Materiales:	Equipos:	
Destornillador de estrella	Brocha, Fanela		
Procedimiento: <ul style="list-style-type: none">- Desconectar el cable de alimentación.- Abrir la tapa frontal.- Limpiar el circuito electrónico con la ayuda de la brocha.- Revisar que los cables estén bien conectados.- Limpiar la superficie interna de la carcasa con la ayuda de una fanela.- Verificar si el los tornillos de sujeción o empotramiento están flojos, ajustarlo si es necesario.- Revisar alguna anomalía en el interior.			
Observaciones: <ul style="list-style-type: none">- Usar guantes para la ejecución de esta tarea			

Fuente: Ejecución de las tareas de mantenimiento de la Facultad de Mecánica.

Limpieza interna. Eliminar cualquier vestigio, desechos, polvo, etc., en las partes internas que componen el módulo, mediante los métodos adecuados según corresponda. Limpieza de tarjetas electrónicas, contactos electrónicos, conectores, utilizando aspirador, brocha, etc.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES.

5.1 Conclusiones

Se diseñó e implementó dos módulos para controlar el ingreso de los usuarios y posterior envío de notificaciones al ingresar al laboratorio de Electrónica para proteger lo que se encuentra en el interior.

Se desarrolló un programa en la plataforma Arduino para interpretar las instrucciones de la red GSM, logrando satisfactoriamente el envío del mensaje de notificación de ingreso de usuarios al laboratorio.

Se grabó correctamente programa al módulo V3.0 por medio de diversas pruebas previas que se realizaron para que controle adecuadamente el acceso del personal encargado al laboratorio de Electrónica.

Se logró la combinación del software LabVIEW y software Arduino para poder controlar un tablero didáctico equipado con sensores digitales y analógicos para que los compañeros puedan conocer las nuevas tecnologías que actualmente innovan al mercado.

5.2 Recomendaciones

Seguir el manual de mantenimiento y el manual de operación para poder preservar los elementos de control que se encuentran en el módulo ya que con el uso adecuado de él se podrá asegurar que los activos físicos que se encuentran en el interior del laboratorio estén seguros es decir que no hay pérdidas que perjudiquen a la enseñanza adecuada a los compañeros.

Encender el módulo en el momento necesario para poder preservar la vida útil de él.

Dar el manejo adecuado al tablero didáctico que está equipado de sensores que se controla por medio del software LabVIEW con comunicación Arduino, y que se pueden

guiar de las características que tiene cada sensor cuyas datos se encuentran en este proyecto con sus respectivas aplicaciones para tener una idea clara de lo que sirve cada uno y para que se utiliza y por ende se pueda ajustar a las necesidades del que lo utilice.

BIBLIOGRAFÍA

Arduino. Arduino.cc. *ArduinoBT*. [En línea] [Citado el: 06 de 07 de 2014.] <http://Arduino.cc/en/pm/wiki.php?n=Guide/ArduinoBT>.

Arduino. 2011. Arduino.cc. *Debido Arduino*. [En línea] 2011. [Citado el: 11 de Julio de 2014.] <http://Arduino.cc/en/pm/wiki.php?n=Main/ArduinoBoardDue>

Bricogeek. 2012. Bricogeek. [En línea] 2012. [Citado el: 10 de Julio de 2014.] <http://tienda.bricogeek.com/cables/343-cable-usb-a-b.html>.

BricoGeek. 2012. Bricogeek.com. *GPS/GPRS/GSM Shield V3.0*. [En línea] 2012. [Citado el: 15 de Julio de 2014.] <http://www.bricogeek.com/shop/shields-Arduino/572-gps-gprs-GSM-shield-V30.html>.

Corto, circuito. 2010. *Introducción al Arduino*. [En línea] 2010. [Citado el: 16 de Agosto de 2014.] <http://www.cortoc.com/2011/12/introduccion-Arduino.html>.

Cyber, WEEK. 2012. *Fotoeléctrico Encoders*. [En línea] 2012. [Citado el: 18 de Agosto de 2014.] http://www.miniinbox.com/es/doble-Velocidad-hc-020k-medicion-modulo-w-fotoelectrico-encoders-negro-Verde-2-pcs_p647629.html.

Diseño, Manufactura. 2011. *Diseño y Manufactura*. [En línea] 2011. [Citado el: 16 de Julio de 2014.] <https://sites.google.com/site/temasdedisenoymanufactura/Arduino>.

DOMINGUEZ, ALBERTO CASTRO. 2000. Alberto Castro Dominguez. [En línea] 2000. [Citado el: 16 de Agosto de 2014.] <https://es.scribd.com/doc/235114033/Pfc-Alberto-Castro-Dominguez>.

Electrónica, CARLOS DÍAZ. 2012. *Sensor de Temperatura de precision*. [En línea] 2012. [Citado el: 20 de Agosto de 2014.] <http://electronica.webcindario.com/componentes/lm35.htm..>

EMBAJADORES, ELECTRÓNICA. 2010. *ARUDINO ATMEGA328* [En línea] 2010. [Citado el: 14 de Julio de 2014.] <http://www.electronicaembajadores.com/datos/pdf1/lc/lca1/lca1001.pdf>.

GPS, Satélites. 2009.. *Satélites GPS* [En línea] 2009. [Citado el: 01 de Julio de 2014.] http://concurso.cnice.mec.es/cnice2006/material121/unidad3/sat_gps.htm.

Kioskea. 2011. *GSM* [En línea] 2011. [Citado el: 01 de Julio de 2014.] <http://es.kioskea.net/contents/681-estandar-GSM-sistema-global-de-comunicaciones-moviles>.

Libre, III Jornada de software. 2004. *Herramientas hardware y software para el desarrollo con microcontroladores PIC*. [En línea] 2004. [Citado el: 02 de Julio de 2014.] <http://www.iearobotics.com/personal/juan/proyectos/skypic-down/skypic.html>.

Libre, Somos. 2004.*Software libre*. [En línea] 2004. [Citado el: 02 de Julio de 2014.] <http://www.somoslibres.org/modules.php?name=Content&pa=showpage&pid=17>.

Manual, de mantenimiento. 2009.*Manual de mantenimiento*. [En línea] 2009. [Citado el: 18 de Agosto de 2014.] <http://es.scribd.com/doc/78037733/Manual-Mantenimiento>.

Manuales. 2014.*GPS*. [En línea] 02 de 07 de 2014. <http://www.manuales.com/manual-de/que-es-gprs>.

National, Instruments. 2012.*LabVIEW*. [En línea] 2012. [Citado el: 16 de Agosto de 2014.] <http://www.ni.com/labview/esa/>.

NEXTIA, FENIX. 2011.*Sensor fotoeléctrico E3F-R2N2*. [En línea] 2011. [Citado el: 18 de Agosto de 2014.] <http://www.nextiafenix.com/product/sensor-e3f-r2n2/>.

Robótica, ELECTAN Electrónica y. 2006.*Arduino UNO*. [En línea] 2006. [Citado el: 14 de Julio de 2014.] <http://www.electan.com/Arduino-uno-con-atmega328-p-2977.html>.

Sensor. 2012.*Sensor de movimiento infrarrojo pir*. [En línea] 2012. [Citado el: 18 de Agosto de 2014.] www.electrobiomedical.com.co/download/datasheet/SEN0013.pdf.

Silicio, M X. 2013.*Sensor de Electricidad*. [En línea] 2013. [Citado el: 18 de Agosto de 2014.] <http://silicio.mx/groVe-sensor-de-electricidad>.

SNOOTLAB. 2011.*Arduino Mega ADK*. [En línea] 2011. [Citado el: 15 de Julio de 2014.] <http://snootlab.com/lang-en/Arduino/296-Arduino-mega-adk-reV-3-en.html>.

Super, Robótica. 2011.*superrobotica*. [En línea] 2011. [Citado el: 17 de Agosto de 2014.] <http://www.superrobotica.com/S320103.htm>.

